

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ**

Proxério Manoel Felisberto

**MECANISMO ONLINE PARA REFERÊNCIAS - MORE:
REENGENHARIA E AMPLIAÇÃO DO SISTEMA**

Araranguá

2013

Proxério Manoel Felisberto

MECANISMO ONLINE PARA REFERÊNCIAS - MORE

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina, como parte dos requisitos necessários para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação.

Araranguá, novembro de 2013.

Proxério Manoel Felisberto

**MECANISMO ONLINE PARA REFERÊNCIAS -
MORE: REENGENHARIA E AMPLIAÇÃO DO SISTEMA**

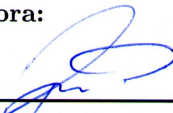
Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Tecnologias da Informação e Comunicação”, e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 28 de novembro 2013.

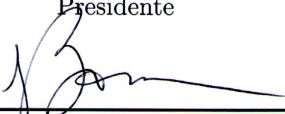


Professor Vilson Gruber, Dr.
Coordenador do Curso

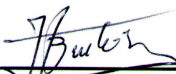
Banca Examinadora:



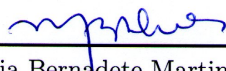
Professor Roderval Marcelino, Dr.
Presidente



Professor João Bosco da Mota Alves, Dr.
Membro



Professor Juarez Bento da Silva, Dr.
Membro



Maria Bernadete Martins Alves, M^a.
Membro

À minha esposa e ao meu filho, cujo apoio
foi fundamental para o êxito desta jornada.

AGRADECIMENTOS

A Deus pela dádiva da vida, por todas as graças recebidas, por minha família e amigos.

A minha família, pela dedicação, incentivo e apoio incondicionais.

A Universidade Federal de Santa Catarina, por sua política de interiorização e pela excelência dos cursos oferecidos à região sulcaterinense.

Ao RexLab Araranguá e seus integrantes, pelo acolhimento e companheirismo durante a realização deste trabalho.

Aos companheiros de caminhada pelas alegrias e angústias compartilhadas.

A todos os docentes, pela sabedoria com que mostraram a trilha que conduz ao conhecimento.

Ao Prof. Dr. Juarez Bento da Silva, pela solicitude e dedicação sempre evidenciadas.

Especialmente, ao Prof. Dr. Roderval Marcelino, pela orientação sempre precisa, pela paciência com meus anseios por vezes equivocados, pelo apoio e confiança demonstrados durante a realização deste empreendimento.

“Conhecer a si próprio é o maior saber.”

Galileu Galilei

RESUMO

Os usuários de biblioteca contam, atualmente, com diversas ferramentas geradoras e/ou gerenciadoras de referências bibliográficas, porém, a maioria delas é proprietária, necessitando a aquisição de algum tipo de licença onerosa para desfrutar de todas as suas possibilidades. Algumas possuem a versão gratuita, no entanto, são muito limitadas e por vezes não contemplam a geração de referências nos moldes especificados pela ABNT. Outras não oferecem a possibilidade de armazenamento das referências geradas para que os usuários as utilize em momento posterior.

Visando suprir esta lacuna foi desenvolvido e disponibilizado ao público em geral através da web, no ano de 2006, o Mecanismo Online para Referências (MORE). Todavia, os avanços tecnológicos exigem que os sistemas web atuais apresentem características que, até um passado recente, não eram essenciais.

Este trabalho visa a adequação do MORE às necessidades atuais da web, fazendo uso, exclusivamente, de tecnologias popularmente consagradas e gratuitas.

Os capítulos iniciais dedicam-se ao levantamento do estado da arte para ferramentas gerenciadoras de referências bibliográficas e as tecnologias gratuitas utilizadas na construção de sistemas web interativos. Os capítulos finais descrevem o processo de reengenharia a que o MORE foi submetido, sua nova estrutura, novas possibilidades implementadas com sua remodelagem e ampliação do seu portfólio de funcionalidades. Finalizando, faz-se uma síntese dos resultados alcançados em contrapartida aos objetivos propostos inicialmente e propõe algumas sugestões para trabalhos futuros visando o aprimoramento do MORE.

Palavras-chave: referências bibliográficas, sistema web, citações, documento jurídico.

ABSTRACT

Currently, library users have several tools that generate and/or manage bibliographical references, but most of them are proprietary, requiring the acquisition of some sort of costly license to enjoy all its features. Some of them have free versions, but they are very limited and do not often include the generation of references as specified by ABNT. Others do not offer the possibility of storing the generated references for users to use them later.

Aiming to fill this gap, the Mechanism for Online References (MORE) has been developed and made available to the general public in the web, back in 2006. However, technological advances requires that nowadays have web systems characteristics that, until the recent past, were not essential.

This work aims to adjust MORE to the current needs of the web, making exclusively use of popularly consecrated and free technologies. The initial chapters are dedicated to the research of state of the art tools for manage bibliographical references and free technologies used to build interactive web systems. The final chapters describes the re-engineering process that MORE was submitted, its new structure, new features implemented from its remodeling and expansion of the existing ones.

Finally, it synthesis the achieved results in contrast to the original objectives and proposes some suggestions for future work aimed to improve MORE.

Keywords: references, web system, citation, legal document.

LISTA DE FIGURAS

Figura 1	O processo <i>Scrum</i>	29
Figura 2	Janela Principal do JabRef®	31
Figura 3	Janela do Microsoft Word TM mostrando o plugin do EndNote® instalado	34
Figura 4	Página inicial do Sistema Facilis.....	35
Figura 5	Página inicial do MORE.....	41
Figura 6	Tela do <i>RefWorks</i> para uma entrada manual de dados .	44
Figura 7	NetBeans IDE 7.1.1	48
Figura 8	Visão geral da arquitetura da plataforma Eclipse	50
Figura 9	Aptana Studio 3.....	52
Figura 10	Astah Community.....	54
Figura 11	O processo de reengenharia.....	67
Figura 12	Diagrama de Casos de Uso do Módulo Suporte	76
Figura 13	Diagrama de Casos de Uso do Módulo Controle de Usuários	76
Figura 14	Diagrama de Casos de Uso do Módulo Controle de Referências	76
Figura 15	Modelo lógico da base de dados do MORE.....	78
Figura 16	Diagrama de Sequência de Referências a Livros.....	79
Figura 17	Diagrama de Comunicação Cadastrar Referência a Livros	80
Figura 18	Diagrama de Máquina de Estado de Cadastro de Referência a Livros	81
Figura 19	Tela do sistema referente ao formulário de contato.	86
Figura 20	Tela do sistema com o formulário de referência a livros.	87
Figura 21	Tela do sistema com os dados de uma pesquisa.....	89
Figura 22	Tela de inclusão e gerenciamento de coleções.....	90
Figura 23	Tela de gerenciamento de referências.....	90
Figura 24	Tela de elaboração de referência a patente.....	92

LISTA DE TABELAS

Tabela 1	Principais sites que usam PHP.....	57
Tabela 2	Requisitos Funcionais do Módulo Suporte	72
Tabela 3	Requisitos Funcionais do Módulo Controle de Usuários	72
Tabela 4	Requisitos Funcionais do Módulo Controle de Referências	72
Tabela 5	Exemplo de Descrição de Caso de Uso	77

LISTA DE ABREVIATURAS E SIGLAS

CWYW	Cite While You Write.....	33
ISI	Institute for Scientific Information	36
RSS	Really Simple Syndication	39
MORE	Mecanismo Online para Referências	41
IDE	Interface Development Environment.....	47
SQL	Structured Query Language.....	48
SGBDR	Sistema de Gerenciamento de Banco de Dados Relacional	48
API	Application Programming Interface	49
FTP	File Transfer Protocol.....	52
UML	Unified Modeling Language	54
PHP	PHP Hypertext Preprocessor.....	55
CGI	Common Gateway Interface.....	56
HTML	HyperText Markup Language	59
HTTP	HiperText Transfer Protocol	61
W3C	World Wide Web Consortium.....	61
CSS	Cascading Style Sheets.....	63
JSP	Java Server Pages.....	68
RF	Requisito Funcional	71
RNF	Requisito Não Funcional	73
IES	Instituição Superior de Ensino	75
MVC	Model, View, Controller.....	82
ABNT	Associação Brasileira de Normas Técnicas.....	86
URL	Uniform Resource Locator	93
XSS	Cross-Site Scripting.....	93
CSRF	Cross-Site Request Forgery.....	93

SUMÁRIO

1 INTRODUÇÃO	25
1.1 OBJETIVOS	26
1.2 JUSTIFICATIVA	27
1.3 METODOLOGIA	28
1.4 ORGANIZAÇÃO DO TRABALHO	30
2 GERENCIADORES DE REFERÊNCIAS BIBLIOGRÁFICAS	31
2.1 JABREF REFERENCE MANAGER	31
2.2 ENDNOTE	32
2.3 FACILIS	35
2.4 PROCITE®	36
2.5 REFERENCE MANAGER®	37
2.6 CITATION MACHINE	39
2.7 EASYBIB	40
2.8 MECANISMO ONLINE PARA REFERÊNCIAS (MORE)	41
2.9 REFWORKS	42
3 TECNOLOGIAS GRATUITAS UTILIZADAS NO DESENVOLVIMENTO DE SISTEMAS WEB	47
3.1 AMBIENTES DE DESENVOLVIMENTO INTEGRADO	47
3.1.1 NetBeans	47
3.1.2 Eclipse	49
3.1.3 Aptana Studio 3	51
3.1.4 Astah Community	53
3.2 PHP	55
3.3 CODEIGNITER	57
3.4 JAVASCRIPT	59
3.5 HTML	61
3.6 CASCADING STYLE SHEETS (CSS)	63
3.7 ZURB FOUNDATION	64
3.8 SGBDR MYSQL	64
4 REENGENHARIA DO MORE	67
4.1 TRADUÇÃO DO CÓDIGO-FONTE	68
4.2 ENGENHARIA REVERSA DO MORE	69
4.2.1 Documento de Requisitos do MORE	69
4.2.1.1 Concepção do Sistema	70
4.2.1.2 Visão Geral do Produto	70
4.2.1.3 Premissas, Restrições e Dependências	71

4.2.1.4	Arquitetura do Sistema	71
4.2.1.5	Requisitos Funcionais	71
4.2.1.6	Requisitos Não Funcionais	73
4.2.1.7	Modelagem dos Casos de Uso	75
4.2.1.8	Modelagem Lógica do Banco de Dados	77
4.2.1.9	Diagramas de Sequência e Comunicação	79
4.2.1.10	Diagrama de Estado	80
4.3	MELHORIA DE ESTRUTURA DE PROGRAMA	82
4.3.1	Model.....	82
4.3.2	View	82
4.3.3	Controller.....	83
4.4	MODULARIZAÇÃO DE PROGRAMA	83
4.5	REENGENHARIA DE DADOS	84
5	RECONSTRUÇÃO E AMPLIAÇÃO DO SISTEMA .	85
5.1	RECONSTRUÇÃO DO MORE.....	85
5.1.1	Módulo de Suporte	85
5.1.2	Módulo de Controle de Referências	86
5.1.3	Módulo de Controle de Usuários	88
5.2	AMPLIAÇÃO DO PORTFÓLIO DE RECURSOS DO SISTEMA	91
6	CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS.....	95
	REFERÊNCIAS	99

1 INTRODUÇÃO

Uma das premissas básicas no tocante à pesquisa científica envolve a utilização de métodos e, consequentemente, técnicas apropriadas para que seus resultados sejam aceitos como cientificamente válidos.

Dentre as técnicas consagradas e amplamente utilizadas destaca-se a utilização de trabalhos já realizados, de preferência por autores reconhecidos no meio científico ou que constem de publicações renomadas na área do saber objeto da pesquisa, para fundamentar ou realizar questionamentos relativos ao tema abordado. Esta técnica requer a concessão dos méritos ao autor, normalmente na forma de citações e referenciamento bibliográfico.

Podemos verificar a existência de vários órgãos, conhecidos mundialmente, que normatizam, segundo suas diretrizes, a forma como essas referências deverão ser utilizadas. Observa-se, também, que um número considerável de países possuem algum tipo de associação, órgão ou agremiação que faz este trabalho em âmbito nacional.

O Brasil, atualmente, conta a Associação Brasileira de Normas Técnicas (ABNT) cuja finalidade precípua é a normalização técnica em todo o território nacional, abrangendo as mais diversas áreas do saber (ABNT, 2013).

A elaboração dos mais diversos tipos de trabalhos realizados no âmbito acadêmico segundo esta normalização tem causado algum transtorno aos seus autores, principalmente na demanda de tempo para inteirar-se das normas e para elaborar os mais diversos tipos de referências bibliográficas utilizadas no projeto em desenvolvimento de acordo com o tipo de fonte consultada.

Para auxiliar os autores neste tipo de empreendimento, surgiu, em 2006, o Mecanismo Online para Referências (MORE) que vem oferecendo, gratuitamente, ao público em geral diversas possibilidades para a elaboração de referências bibliográficas, conforme informações constantes em sua página eletrônica, no seguinte endereço:

- *<<http://www.rezlab.ufsc.br:8080/more/informacoes.jsp>>*

Os usuários da web dispõem de algumas ferramentas que os auxiliam na elaboração e no gerenciamento de suas referências bibliográficas, porém o MORE visa auxiliar àqueles que buscam uma ferramenta gratuita, que possam armazenar suas referências e que ofereça este serviço segundo as normas da ABNT.

A popularização da internet, aliada aos avanços tecnológicos da

última década, trouxeram consigo algumas exigências que os sistemas web devem cumprir, obrigando-os, muitas vezes, à algum tipo de atualização para que evidenciem, cada vez mais, elevado grau de escalabilidade, robustez, confiabilidade, confienciabilidade, rapidez e interatividade.

O fato é que o MORE está inserido neste contexto e vem apresentando alguns problemas, muitas vezes relatados por alguns de seus quase 90.000 usuários cadastrados, além do que sua manutenção tornou-se difícil e dispendiosa.

Assim, após analisada cuidadosamente a situação em que o MORE se encontra, foi verificada a necessidade de sua adaptação à nova realidade vivenciada.

Do exposto e fazendo uso dos ensinamentos de Loudon (2010) buscar-se-á que o MORE ao findar este trabalho evidencie as características apresentadas pelas grandes aplicações web.

No tocante ao gerenciamento da complexidade, que ofereça disponibilidade contínua, gerencie até 30 (trinta) conexões simultâneas, apresente ao usuário apenas a página solicitada, aumente sua longevidade, suporte atualizações em tempo real e possa ser utilizado pelos principais navegadores.

No que tange a modularidade, que implemente uma camada de abstração de dados a fim de ocultar a forma como o sistema executa a conexão com a base de dados e implemente acoplamentos fracos entre as camadas para que a dependência entre elas seja a mínima possível.

1.1 OBJETIVOS

Esta sessão apresenta o objetivo geral e os objetivos específicos deste trabalho.

- **Geral**

Realizar uma reengenharia e ampliar o portfólio de recursos do Mecanismo Online para Referências.

- **Específicos**

1. Estudar as tecnologias atualmente mais indicadas para o desenvolvimento de sistemas web interativos, as quais deverão ser empregadas na rescrita do código-fonte do MORE, a fim de torná-lo mais eficiente.

2. Remodelar o MORE, de forma a cumprir os requisitos desejados.
3. Desenvolver e implementar uma lógica no sistema de busca, de forma a encontrar qualquer referência armazenada, informando no mínimo 5 (cinco) caracteres, de qualquer um dos campos de qualquer formulário armazenado.
4. Oferecer aos usuários de biblioteca que não disponham de uma ferramenta proprietária a possibilidade de fazer uso de um gerador/gerenciador de referências bibliográficas gratuitamente.
5. Disseminar os conhecimentos adquiridos e promover os sistemas gratuitos.

1.2 JUSTIFICATIVA

Os usuários de bibliotecas encontram, atualmente, no mercado vários aplicativos que permitem o gerenciamento de referências bibliográficas, porém, em sua grande maioria são aplicativos proprietários e requerem algum tipo de customização para que este gerenciamento se faça de acordo com as normas da ABNT, e o acesso a estas normas é feito de forma onerosa.

O Mecanismo Online para Referências (MORE) oferece esta possibilidade de forma gratuita e conta atualmente com quase 90.000 usuários cadastrados e algo próximo a 3.000.000 (três milhões) de referências armazenadas em sua base de dados. Sua página possui 4.000 (quatro mil) visitas diárias em média.

Este ano, apesar do sistema estar apresentando uma série de problemas, foram recebidas solicitações de várias instituições no sentido de autorizar a colocação em suas páginas eletrônicas de um enlace que direcione ao MORE.

O mecanismo vem apresentando, sistematicamente, falhas na comunicação entre o *front-end* e a base de dados. A título de exemplo pode-se citar diversas mensagens reportando que o usuário faz a inserção dos dados e o sistema gera uma referência à outra obra que não aquela cujos dados foram informados. Ou ainda, muitos usuários angustiados por não estarem conseguindo recuperar sua coleção de referências. Outro problema que vem apresentando é a não validação da autenticação de usuários já cadastrados.

Foi levantado, junto a equipe do RexLab, que infelizmente não se dispõe de recursos específicos (e todas as tentativas de obtenção até

agora foram frustradas) para a manutenção do sistema. Assim, faz mais de dois anos que não sofre qualquer tipo de atualização o que vem refletindo na quantidade de problemas apresentados.

Salienta-se as negativas recebidas sobre a disponibilização de recursos tecnológicos para dar suporte aos experimentos e projetos desenvolvidos no RexLab.

No entanto, o grande incentivo para fazer uma reengenharia no MORE e ampliar seu portfólio de recursos advém da considerável quantidade de mensagens solicitando esta ampliação e agradecendo o auxílio prestado pela ferramenta no momento em que se deparam com a necessidade de formatar a lista das referências utilizadas, principalmente, em um determinado trabalho acadêmico.

Para atender aos anseios desta parcela da sociedade, mantendo sua disponibilização de forma gratuita a qualquer um que venha fazer uso de seus serviços é que o MORE necessita passar por uma reengenharia e ampliação de seus serviços.

Este projeto será desenvolvido com a utilização de tecnologias gratuitas que possibilitará a disseminação de paradigmas atuais de desenvolvimento para a web, tais como: HTML, CSS, PHP e JavaScript, entre outros.

1.3 METODOLOGIA

A pesquisa desenvolver-se-á segundo os preceitos da Metodologia da Experimentação, que para Cervo, Bervian e Silva (2007) “é o conjunto de processos utilizados para verificar as hipóteses [...] e o princípio geral no qual se fundamentam as técnicas da experimentação é o do determinismo, que se anuncia assim: nas mesmas circunstâncias, as mesmas causas produzem os mesmos efeitos”.

O desenvolvimento do sistema dar-se-á com a utilização de métodos ágeis, mais precisamente o *SCRUM*. Esta metodologia baseia o desenvolvimento do software nos princípios ágeis e no Manifesto Ágil, ocorrendo de forma iterativa, através de uma série de *sprints*. Um *sprint* pode ser entendido como um lapso de tempo, com duração entre duas e quatro semanas, que são utilizadas para implementar alguma funcionalidade no software e entregá-lo, ainda que incompleto, para ser utilizado e sofrer as críticas necessárias ao seu melhoramento.

Vale salientar que um ciclo de *sprint* é definido por objetivos que são entendidos como etapa isolada do produto como um todo, permitindo que a equipe permaneça concentrada a fim de obter o êxito

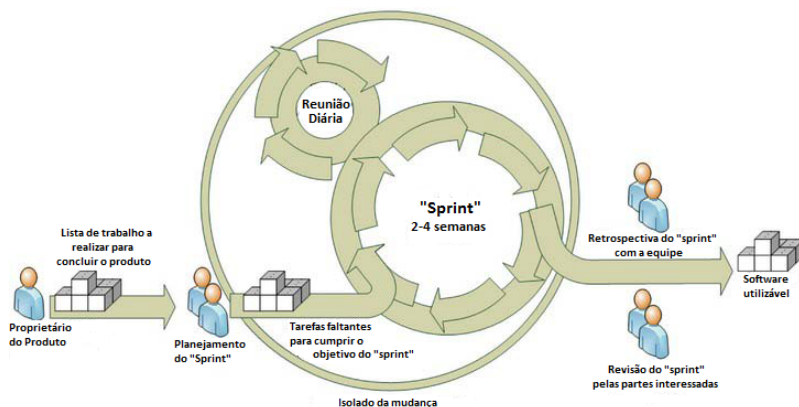
esperado para aquele período de tempo e faça a entrega de um software utilizável e de alta qualidade.

Ao término de um ciclo de *sprint* as tarefas que foram implementadas neste *release* do software são retiradas da lista de trabalho a realizar para a conclusão do produto. Esta lista enumera todas as ações a serem realizadas para que o produto possa ser considerado concluído (em plenas condições de uso) e, se necessário, retroalimentadas com as críticas recebidas.

Assim, para que o trabalho se desenvolva em um ambiente colaborativo, os interessados reúnem-se diariamente e ao final de cada *sprint* para avaliar o andamento das tarefas e acordar pequenos detalhes para a melhoria do produto. Para Blankenship, Bussa e Millett (2012) o “aspecto iterativo do *Scrum* é que este ciclo é repetido várias vezes até que o projeto esteja completo”.

A Figura 1 mostra graficamente as etapas do processo de construção de um software utilizando-se a metodologia *Scrum*.

Figura 1 – O processo *Scrum*.



Fonte: adaptada de Blankenship, Bussa e Millett (2012).

As tarefas abaixo farão parte do desenvolvimento deste trabalho:

- Elaborar o documento de requisitos do MORE, de modo a facilitar sua reengenharia e, posterior, manutenção.
- Implementar novos formulários de forma a contemplar a elaboração de referências aos documentos jurídicos.

- Implementar a possibilidade de importação de uma referência armazenada na base de dados para uma coleção pessoal.
- Implementar a possibilidade de um usuário possuir mais de uma coleção.
- Repaginar visualmente o MORE, porém buscando manter suas características visuais.

1.4 ORGANIZAÇÃO DO TRABALHO

Este documento, além desta introdução, está organizado em mais 5 (cinco) capítulos que abordam os seguintes conteúdos:

O **Capítulo 2** faz um apanhado sobre os principais gerenciadores de referências bibliográficas disponíveis aos usuários de bibliotecas, porém alguns são proprietários e requerem a aquisição de algum tipo de licença para a sua plena utilização.

O **Capítulo 3** descreve algumas definições e características de várias tecnologias empregadas, atualmente, no desenvolvimento de sistemas para a web. Aborda apenas as tecnologias disponibilizadas gratuitamente, uma vez que as proprietárias não se coadunam com o escopo deste trabalho.

O **Capítulo 4** traz uma descrição de todo o processo desenvolvido na reengenharia do MORE. Iniciando com a tradução do código-fonte, passando pela engenharia reversa cujo principal produto foi o documento de requisitos, além da melhoria da estrutura do sistema, modularização deste e reengenharia dos dados.

O **Capítulo 5** aborda a sequência de tarefas utilizadas na implementação, testes e entregas, que viabilizaram a reconstrução do MORE e a ampliação do seu portfólio de recursos.

O **Capítulo 6** busca fazer um apanhado dos resultados obtidos a fim de verificar se os objetivos propostos foram alcançados e indicar possibilidades de melhorias a serem consideradas em trabalhos futuros.

2 GERENCIADORES DE REFERÊNCIAS BIBLIOGRÁFICAS

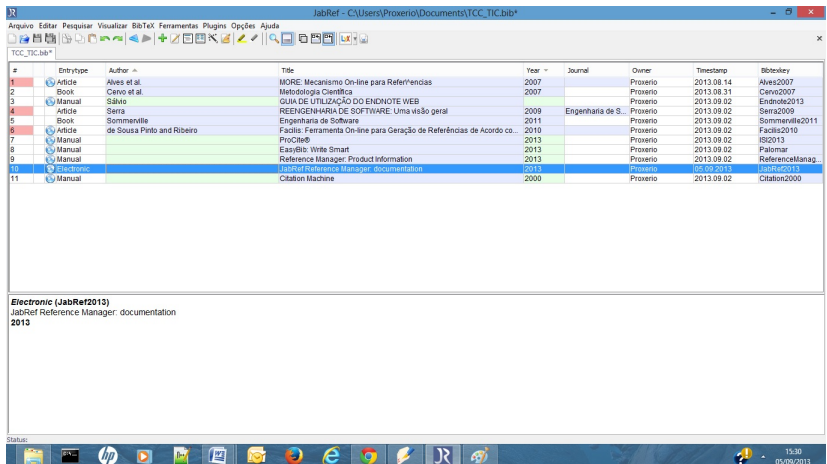
Este capítulo fará um apanhado sobre os principais gerenciadores de referências bibliográficas disponíveis aos usuários de bibliotecas, porém alguns são proprietários e requerem a aquisição de algum tipo de licença para a sua plena utilização.

2.1 JABREF REFERENCE MANAGER

O JabRef[®] é um gerenciador de referências gratuito, desenvolvido em Java, cuja execução se dá através da Máquina Virtual Java 1.5 ou superior, o que lhe confere a portabilidade para várias plataformas.

A Figura 2 mostra uma vista da janela principal do aplicativo em execução. Segundo SOURCEFORGE.NET (2013), apesar de exportar

Figura 2 – Janela Principal do JabRef[®]



Fonte: print screen da aplicação no sistema operacional Windows 8.

suas referências para vários tipos de arquivos, inclusive para o editor de textos Microsoft WordTM, sua característica principal é usar nativamente o formato de arquivo BibTex, o mesmo utilizado pelo L^AT_EX para gerenciar as referências bibliográficas. Cabe esclarecer que BibTex

é um aplicativo que viabiliza o processamento automático de referências bibliográficas nos documentos \LaTeX e no CiteMaker da Adobe[®] FrameMaker.

Possibilita ao usuário a personalização dos tipos de referências, inclusive a inclusão, exclusão ou alteração de campos, sejam eles obrigatórios, opcionais ou de uso geral. A base de dados gerada no JabRef[®] pode ser acessada diretamente pelo \LaTeX bastando apenas informar o caminho do arquivo BibTex. Para ter-se uma visão geral do conteúdo da base de dados basta fazer uso da funcionalidade de classificação e pesquisa.

Dispõe do recurso de gravação automática provendo, desta maneira, a recuperação dos dados na ocorrência de uma falha no computador ou no aplicativo.

Oferece, também, a possibilidade de gerenciamento de grupos, cuja finalidade é o compartilhamento de dados entre vários usuários. Estes grupos podem ser tipificados da seguinte forma: grupo de todas as entradas (não pode ser editado ou excluído); grupos estáticos (contém somente as entradas especificadas pelos usuários); grupos dinâmicos baseados em palavras-chave; e, grupos dinâmicos baseados em expressões de busca de forma livre (inclusive expressões regulares).

O usuário tem a seu dispor para a realização de pesquisas as seguintes bases distribuídas pela web: ACM Portal, ADS from ADS-DOI, ArXiv.org, CiteSeerX, DBLP, DOI to BibTex, DiVA, Google Scholar, IEEEExplore, INSPIRE, ISBN to BibTex, JSTOR, Medline e SPIRES.

Para a utilização do JabRef[®] é conveniente que o usuário possua alguma familiaridade com o manuseio de aplicativos em geral, uma vez que requer a customização do ambiente, bem como a instalação de plugins para ativação de algumas funcionalidades. Por exemplo, sua utilização com o editor de textos Microsoft WordTM.

2.2 ENDNOTE

Os assinantes da plataforma ISI Web of Knowledge (Web of Science) contam com uma ferramenta desenvolvida pela Thomson Reuters, o EndNote Web[®]. Segundo Pestana e Funaro (2009) trata-se de “um software de gestão pessoal que auxilia o aluno de graduação, pós-graduação, docente e pesquisador na elaboração de artigos de pesquisa”. E para Estorniolo Filho (2013) “permite armazenar e organizar as referências encontradas nas buscas em bases de dados, possibilitando a inclusão automática de citações e referências quando da elaboração

de um texto”. É o gerenciador de referências preferido dos usuários do editor de textos Microsoft WordTM.

Na realidade o EndNote[®] dispõe de três formas de utilização. O mais complexo e que apresenta maior número de funcionalidades é o EndNote[®] para desktop. Temos ainda o EndNote Web[®] para assinantes e o EndNote Web[®] versão livre. Este porém com pouquíssimos recursos, se comparado aos demais.

Sálvio (2013) apresenta algumas funcionalidades que podem ser encontradas no aplicativo, dependendo do tipo utilizado, as quais são listadas na sequência:

- Oferece a possibilidade de geração de referências bibliográficas em mais de 600 estilos de normas, quer sejam nacionais ou internacionais;
- A partir do EndNote Web é possível realizar pesquisas em várias bases de dados, dentre elas, Web of Science e PubMed, cujo resultado poderá ser adicionado automaticamente à uma coleção existente ou criar uma nova a partir do resultado obtido;
- Localização e exclusão de registros duplicados;
- Compartilhamento de grupos de registros com outros usuários, podendo ser estabelecido níveis, tais como: somente leitura, leitura e edição, ou mesmo, exclusão;
- Oferece a possibilidade de armazenagem de até 5GB de anexos (PDF e imagem), além de até 10.000 registros em 5.000 grupos.
- Com a utilização do plugin “Cite While You Write” (CWYW) é possível fazer a inserção automática de citações e a formatação de referências bibliográficas no Microsoft WordTM;

A Figura 3 mostra uma vista da janela principal do aplicativo em execução.

É conveniente, ao término do trabalho, fazer uma verificação na lista de referências e nas citações a fim de localizar possíveis divergências com a norma escolhida, uma vez que os dados disponíveis nas bases não se encontram no mesmo padrão e a variedade de normas trabalhadas é considerável.

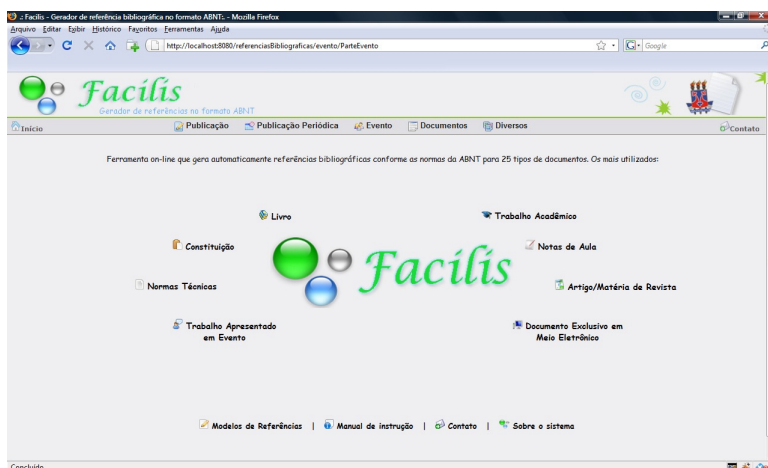
Após o usuário identificar-se o aplicativo mostra um menu com 5 (cinco) opções: My References, Collect, Organize, Format e Options. Em “My References” é possível ter acesso a todos os registros cadastrados ou distribuídos pelos grupos. Com “Collect” escolhe-se uma das

2.3 FACILIS

Segundo Pinto e Ribeiro (2010) o “Facilis é um ferramenta gratuita que proporciona aos seus usuários a geração de referências conforme os padrões estabelecidos pela Associação Brasileira de Normas Técnicas (ABNT), sem que eles tenham que se preocupar com essas normas. Dessa maneira ela diminui as dificuldades de redigir referências teóricas, facilitando e agilizando a conclusão do trabalho científico”.

A Figura 4 traz uma vista da página inicial do sistema na ocasião do acesso à sua página eletrônica.

Figura 4 – Página inicial do Sistema Facilis



Fonte: Pinto e Ribeiro (2010).

Oferece a geração de referências bibliográficas através da inserção manual dos dados e o idioma utilizado é o português brasileiro. Está disponível no endereço eletrônico <<http://facilis.uesb.br>>, requerendo do usuário apenas a utilização de um navegador web e, é claro, conexão à rede mundial de computadores. Se propõe a gerar todos os tipos de referências constantes na NBR 6023 (2002), porém não oferece a possibilidade de gerenciamento destas através de algum dispositivo de armazenamento e não faz nenhuma alusão à NBR 10520 (2002), ou seja, não auxilia seu usuário no uso de citações. A fim de atender aos princípios ergonômicos, dispõe de links de ajuda, uma página com modelos de referências, um pequeno manual de instruções e marcação

de saídas.

2.4 PROCITE®

O ProCite® é uma ferramenta proprietária de gerenciamento de documentos e referências bibliográficas do *Institute for Scientific Information (ISI)*.

Universidad de Sevilla (2006) o define da seguinte forma (texto traduzido do original em espanhol):

ProCite é um programa de gestão bibliográfica, útil para organizar referências e criar bibliografias automaticamente a partir de um processador de textos. Permite construir bases de dados com as referências bibliográficas recuperadas através das pesquisas realizadas em qualquer suporte. É um dos gestores de bases de dados mais utilizados. É uma ferramenta desenhada para criar, manter, organizar e dar forma à referências bibliográficas ou citações de artigos de revista ou livros, obtidas de uma ou de várias bases de dados. Nos ajuda a pesquisar, descarregar, organizar e apresentar essas referências bibliográficas no estilo desejado.

A efetividade para a gestão de referências bibliográficas fica plenamente evidenciada na ocasião em que o usuário faz a importação diretamente de seu processador de textos, quer seja para inserir uma citação no texto ou uma referência em sua lista de referências, além de manter uma base de dados em linha ou editá-la em algum meio de armazenamento físico. Esta funcionalidade é adicionada ao processador de textos com a instalação do plugin *Cite While You Write (CWYW)* (E-PRINTS, 2006).

Trata-se de um sistema de banco de dados de propósito específico uma vez que não necessita nenhuma programação para gerenciar grandes bases de dados com registros de tamanhos variáveis. Trabalha com o conceito de *workforms* que oferece ao usuário uma espécie de modelo contendo as entradas e as saídas já formatadas para os mais diversos tipos de documentos, tais como: livros, monografias, artigos de periódicos, fotografias, filmes, patentes e mapas, entre outros.

As funcionalidades a seguir são salientadas por ISI RESEARCH-SOFT (1999):

- Quantidade e tamanho das bases de dados limitada somente pela

capacidade de armazenamento;

- Conexão com bases de dados externas através da internet;
- Pesquisas complexas com a utilização de operadores booleanos e/ou relacionais, sem exigir do usuário conhecimento avançado sobre recuperação de informação;
- Possibilidade de armazenagem das expressões de buscas utilizadas;
- Edição das bibliografias obtidas através de pesquisas para estilos conhecidos ou, inclusive, personalizá-los;
- Possibilita a criação e o gerenciamento de grupos de registros, para tratá-los de forma independente;
- Permite um controle sobre autores e títulos, possibilitando a pesquisa e a substituição, eliminando registros duplicados;
- Dispõe da funcionalidade *bibliolink* o que permite gerenciar base de dados pessoais geradas por, ou para, outros aplicativos semelhantes.

2.5 REFERENCE MANAGER[®]

O Reference Manager[®] destaca-se como uma das ferramentas proprietárias da Thomson Reuters para gerenciamento de referências, uma vez que importa referências bibliográficas de bases de dados remotas; permite organizar com facilidade uma coleção pessoal de referências e sua publicação na web ou em uma intranet; disponibiliza o plugin “Cite While You Write” para utilização no MS WordTM ou Corel[®] WordPerfect; permite o compartilhamento em rede da coleção de referências. (THOMSON REUTERS, 2013).

A limitação da quantidade de bases de dados e da informação nelas armazenadas é estabelecida pelo equipamento utilizado, porém é escalável, uma vez que estas bases podem ser compartilhadas em rede. As informações são armazenadas em campos de tamanho variável oferecendo grande flexibilidade quanto ao tipo de informação a ser armazenada. Logicamente a geração das referências e citações está diretamente relacionada à maneira como estas informações são inseridas. Em sua versão padrão, disponibiliza 35 (trinta e cinco) tipos de estilos de formatação, cada um com até 37 (trinta e sete) campos diferentes,

os quais podem estar situados na mesma base de dados. Ao usuário é permitido modificar qualquer um dos tipos predefinidos, porém esta funcionalidade deve ser utilizada com cautela.

Existem várias formas que poderão ser utilizadas para a inserção de um registro na base de dados. Estes registros podem ser importados em linha, diretamente do *Reference Manager*, através do acesso ao *Web of Knowledge*, *PubMed* ou qualquer uma das centenas de bases de dados que utilizam o protocolo Z39.50. Também podem ser importados de um arquivo tipo *RIS*, de uma base de dados pessoal ou de um dispositivo que armazenamento físico que contenha alguma base de dados compatível. O conjunto de referências advindas de uma pesquisa poderá ser armazenado em uma nova base de dados. Esta nova base de dados deverá receber um nome distinto das demais. Ou então, poderá selecionar uma ou mais destas referências e arrastá-las à uma base de dados existente. Lembrar que as pesquisas devem conter termos de busca que definam bons filtros porque pesquisas genéricas poderão retornar um conjunto de milhares de registros. (THOMSON REUTERS, 2008).

A utilização da funcionalidade de importação de referências pode acabar gerando alguma referência com pequenos erros os quais poderão ser corrigidos. Através da edição pode-se modificar ou introduzir dados nos mais diversos campos. Recomenda-se verificar a existência de registros duplicados, esta é uma das funcionalidades do *Reference Manager*, e eliminá-los.

Uma base de dados organizada oferece a possibilidade de exportação de referências para serem utilizadas por outros gerenciadores de referências bibliográficas, por uma tabela do MS ExcelTM, etc. Segundo nos ensina Universidad Nacional de Colombia (2006) o *Reference Manager* realiza a exportação nos formatos:

- **RIS** - é um formato utilizado para intercâmbio entre alguns aplicativos bibliográficos que mantém a formatação da fonte, tais como, letra em negrito ou cursiva.
- **MEDLARS** - é um formato de arquivo de texto utilizado pela *National Library of Medicine*.
- **COMMA DELIMITED** - é um formato que faz a separação dos campos através da utilização de vírgula. De grande utilidade para as bases de dados SQL.
- **TAB DELIMITED** - este formato utiliza-se de tabulações para realizar a separação dos campos de informação. Pode ser utilizado

por aplicativos de folhas de cálculo. Por exemplo, o MS ExcelTM.

- XML - formato amplamente utilizado na web, por exemplo, por sítios de notícias que oferecem *Feeds RSS*.

Dispõe do plugin *Cite While You Write (CWYW)* que uma vez instalado no editor de textos disponibiliza, neste, uma barra de ferramentas com as seguintes opções: *Insert Citation*, *Go to Reference Manager*, *Insert Marked Reference(s)*, *Instant Formatting*, *Generate Bibliography*, *Edit Citation(s)*, *Edit Database Reference(s)*, *Revert to Original Text*, *Remove Field Codes*, *Export Travelling Library*, *Cite While You Write Preferences* e *Help*. Entre outras possibilidades, com a utilização do CWYW é possível inserir uma citação diretamente no texto e automaticamente a referência será adicionada à lista, de acordo com modelo escolhido.

2.6 CITATION MACHINE

Segundo THE LANDMARK PROJECT (2013) o *Citation Machine* tem por objetivo auxiliar estudantes e pesquisadores profissionais na elaboração de referências bibliográficas, fazendo com que esta tarefa se torne um procedimento fácil. Elaborado originalmente em 2000, por David Warlick, no âmbito do Projeto Landmark. Oferece a possibilidade de utilização nos formatos MLA, APA, TURABIAN e CHICAGO.

O sistema é gratuito e interativo. Encontra-se disponível na web e não oferece a possibilidade de customização para outro formato além dos 4 (quatro) já mencionados.

Para gerar uma referência o usuário fará a escolha do formato desejado através de um menu disponível no lado esquerdo da página. Na sequência deverá escolher um dos seguintes tipos de fontes: página web, livro, trabalho compilado, obras de referência (enciclopédia), publicação governamental, conferência, trabalho acadêmico, jornal (artigo), revista (artigo), A/V mídia, TV/Rádio, web imagem, blog, debate on-line, *podcast* ou entrevista. Neste momento será apresentado ao usuário os campos a serem preenchidos manualmente, de acordo com a sua opção. Após o preenchimento dos campos a referência e a citação poderão ser geradas segundo o formato previamente escolhido.

Não dispõe de uma base de dados para o armazenamento das referências geradas e não oferece a possibilidade de exportação para qualquer tipo de arquivo.

2.7 EASYBIB

Para PALOMAR COLLEGE LIBRARY (2013) a “*EasyBib* é uma ferramenta online que ajuda a criar citações nos estilos MLA, APA e Chicago/Turabian. Operacionaliza a criação de uma lista de obras referenciadas. Assim, em muitos casos, *EasyBib* pode criar uma referência automaticamente (AutoCITE) se reconhecer o livro, site, artigo de revista, etc, que está sendo citado”.

Apresenta os 6 (três) níveis de usuários a seguir mencionados: *EasyBib free*, *EasyBib Mybib Pro*, *EasyBib School Edition*, *Library Edition*, *EasyBib for iPhone* e *EasyBib for Google Apps*. Dos níveis apresentados, apenas o *EasyBib free* é gratuito e seus recursos limitados a apenas gerar referências no formato MLA e possibilitar seu armazenamento na base de dados do aplicativo, se possuir uma conta (registro) e estiver autenticado. Todos os demais recursos são considerados premium e de exclusividade dos assinantes dos outros cinco níveis.

Para gerar uma referência acessa-se a aba *Bibliography*, escolhe um dos três estilos pré-formatados e um dos 59 (cinquenta e nove) tipos de fonte. O sistema apresentará os campos para preenchimento, conforme os parâmetros estipulados. Se após o preenchimento dos campos o usuário optar por gerar a referência, esta lhe será apresentada juntamente com a opção para armazená-la na base de dados.

A funcionalidade de importação de referências localizadas em bases de bases de dados externas é facultado somente aos usuários que tenham assinatura paga. O *EasyBib* possibilita a importação de referências das bases dados de *ProQuest*, *JSTOR*, *RefWorks* e EBSCO, entre outras e a exportação para arquivos tipo ABC-CLIO.

Uma referência armazenada na base de dados poderá ser consultada e, através do botão “Exportar ou Imprimir”, o usuário poderá escolher uma das 5 (cinco) opções apresentadas: “Imprimir um Documento Word”, “Guardar no Google Docs”, “Guardar no SkyDrive”, “Enviar por E-mail” e “Copiar & Colar”.

Além de guardar, importar e exportar referências, permite a criação de várias coleções por usuário, como também, o compartilhamento com outros usuários. Assim, se estiver autenticado, ao efetuar um clique no botão *Sharing*, 3 (três) opções são exibidas: enviar por e-mail, convidar outros e compartilhar link público.

O compartilhamento pode ser configurado de maneiras diferentes para cada coleção (grupo), assim uma coleção poderá ser compartilhada para somente leitura (os outros usuários não podem alterar os dados da referência) e outra coleção do mesmo usuário para escrita e leitura

(compartilhamento total).

2.8 MECANISMO ONLINE PARA REFERÊNCIAS (MORE)

O Mecanismo Online para Referências oferece, gratuitamente, ao público em geral, diversas possibilidades para a elaboração de referências bibliográficas, conforme nos esclarece Alves, Mendes e Alves (2007):

O MORE é uma ferramenta gratuita e fácil de usar, que produz automaticamente citações no texto e referências no formato ABNT, para quinze (15) tipos de documentos, a partir de formulários próprios, selecionados em um menu principal. Os documentos cobertos pelo mecanismo são os mais usados no meio acadêmico: livros, dicionários, enciclopédias, teses e dissertações, artigos de revistas, artigos de jornais, nos formatos impresso e eletrônico, além dos documentos exclusivos em meio eletrônico: homepage e e-mail.

O MORE apresenta uma interface intuitiva e amigável, cuja página inicial é mostrada na Figura 5. Todo o desenvolvimento do

Figura 5 – Página inicial do MORE



Fonte: <http://www.rexlab.ufsc.br:8080/more/>.

MORE está baseado na utilização de ferramentas disponibilizadas como softwares livres, não proprietárias, amplamente utilizadas por toda a comunidade de desenvolvimento de softwares, bem sedimentadas e com atualizações frequentes.

O MORE atende um público bem específico, ou seja, aquele que necessita uma ferramenta de gerenciamento de referências bibliográficas gratuita, em português, que gere suas referências de acordo com a NBR 6023 (2002), que não está disposto pagar pelo uso de uma ferramenta proprietária, que sua instituição não disponibiliza alguma ferramenta do gênero e que possa manter suas referências armazenadas e disponíveis na web para serem acessadas a qualquer momento e de qualquer lugar.

Esta ferramenta conta, hoje, com mais de 90.000 (noventa) mil usuários cadastrados e sua base de dados, atualmente, possui em torno de 3.000.000 (três milhões) de referências armazenadas. Sua página web recebe quase 4.000 (quatro) mil visitas diariamente.

2.9 REFWORKS

Esta seção abordará uma ferramenta de gerenciamento de referências bibliográficas e de base de dados, proprietária e disponível na web. Trata-se do RefWorks da ProQuest.

Este sistema web, segundo Universidad Rey Juan Carlos (2007),

É uma ferramenta de gerenciamento de referências bibliográficas que permite importar referências de diversas fontes de informação (revistas, livros, catálogos de bibliotecas, provedores de revistas eletrônicas, páginas web, etc...), gerenciar sua própria base de dados pessoal de referências bibliográficas, formatar bibliografias e documentos nos estilos profissionais e compartilhar informações.

Possui dois tipos de assinantes, o individual e o institucional. Assim, para fazer uso dos serviços oferecidos pelo *Refworks* é necessário que o usuário disponha de uma assinatura paga ou esteja incluído em um grupo, cujos serviços serão indenizados pela organização que gerencia o grupo. Para a autenticação do usuário de uma assinatura coletiva faz-se necessário que este informe a chave do grupo a que pertence ou utilize-se de um IP incluso na faixa de IPs cadastrados para a sua organização.

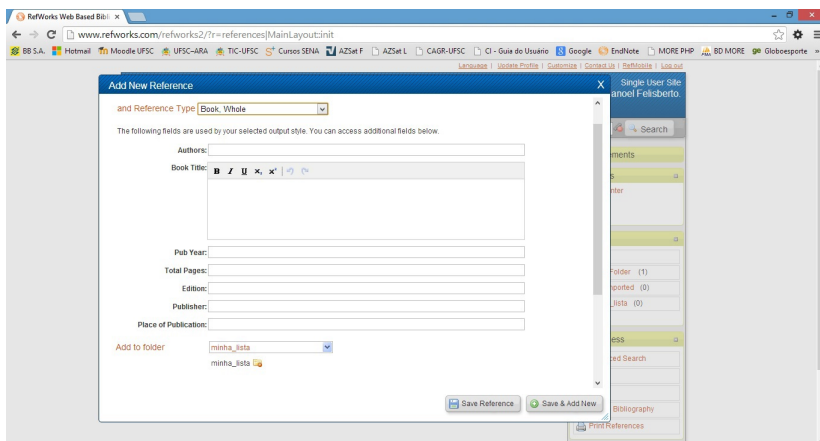
Suas funcionalidades, para efeito didático, serão divididas em 6 (seis) grupos: entrada de dados, organização dos registros, lista de referências bibliografias, pesquisa, edição da informação e *RefShare*. Para dar suporte a estas funcionalidades e melhorar a acessibilidade, dispõe dos seguintes recursos avançados, os quais foram adaptados de ProQuest (2013b):

- *Write-N-Cite* - oferece, ao usuário, a possibilidade de inserção de uma citação diretamente no texto;
- *RefGrab-It* - para localizar e importar informações bibliográficas a partir de uma página da web;
- *Attachment Feature* - para anexar arquivos de vários tipos, desde que não ultrapassem 20MB, às referências constantes de sua base de dados;
- *RefMobile* - é uma interface para dispositivos móveis;
- *RefShare* - possibilita o compartilhamento de suas referências com outras pessoas;
- *Scopus Edition* - versão do *Refworks* que possibilita a interoperabilidade entre ele e o *Scopus*.

A entrada de dados para uma base pessoal no *RefWorks* poderá ocorrer de 6 (seis) formas diferentes. Diretamente de alguma base de dados disponível na internet, tais como: CSA, OCLC, Ebsco, ProQuest, Ovid, ScienceDirect e Google Scholar, entre outras. De um arquivo de texto, facilitada pela existência de filtros de importação para uma grande quantidade de formatos, provedores e programas. Através de uma entrada manual, onde o usuário deverá escolher um formato e o tipo da fonte para que o sistema lhe indique quais campos são de preenchimento obrigatório, além conduzi-lo durante o processo de realização da tarefa. De catálogos bibliográficos que utilizem o protocolo Z39.50. Poderá, também, importar dados de *Feeds RSS*, que atualmente se destaca como forma de apresentação de informações que necessitem atualizações constantes. Por último, a referência bibliográfica poderá ser importada, a partir de uma página web, utilizando-se o recurso *RefGrab-It*, o qual deverá estar instalado na máquina importadora da referência. (UNIVERSIDAD AUTÓNOMA DE MADRID, 2008).

Na Figura 6 temos a tela do *RefWorks* para uma entrada manual de dados. Os registros poderão ser organizados por coleções de

Figura 6 – Tela do *RefWorks* para uma entrada manual de dados



Fonte: <http://www.refworks.com/refworks2/?r=references>.

referências, de acordo com as necessidades do usuário. Não há limite para o número de coleções que cada usuário pode criar.

Uma lista de referências bibliográficas personalizada poderá ser criada manualmente, de maneira fácil e rápida, a partir dos registros existentes na base de dados pessoal. Ou ainda, automaticamente, uma vez que, ao inserir citações bibliográficas no documento, o *RefWorks* cria a lista de referências bibliográficas no final deste. Este recurso mostra-se muito útil, para pesquisadores, autores, professores e alunos, na automatização da criação das listas de referências bibliográficas em seus trabalhos, tais como, artigos, teses, dissertações, monografias, etc. (PROQUEST, 2013a).

O *RefWorks* efetua 3 (três) tipos distintos de pesquisa: a rápida, a avançada e através de índices. Na pesquisa rápida será realizada uma busca em todos os campos, a fim de localizar alguma informação, ou parte dela, que coincida com os termos pesquisados. A pesquisa avançada baseia-se em filtros refinados, inclusive com a utilização de operadores booleanos, de modo a direcionar e limitar a quantidade de referências resultantes da busca. Finalmente, o usuário poderá valer-se dos índices de autores, de descritores ou de revistas para realizar a pesquisa através de índices, o que facilita, agiliza e reduz o tempo da busca.

Dotado de uma série de funcionalidades, o *RefWorks* possibilita

a edição da informação importada ou armazenada na base de dados. Essas possibilidades, segundo ProQuest (2013a), “permitem organizar, editar, encontrar e visualizar as referências bibliográficas de sua base de dados”. Ao editar uma referência é possível alterar o valor dos campos e, inclusive, anexar um ou mais arquivos à referência editada. Pode-se, também, criar pastas e subpastas que serão utilizadas para o armazenamento das coleções do usuário. A exclusão acontecerá mediante confirmação, evitando exclusões acidentais, uma vez que esta opção elimina, completamente, o registro da base de dados.

O *RefShare* é um módulo do *RefWorks* que permite o compartilhamento da base de dados com usuarios da sua instituição ou fora dela. Para Universidad de Salamanca (2006) “os usuários de *RefShare* podem exportar, imprimir, gerar listas de referências e, inclusive, personalizar o formato de saída, mas nunca estarão autorizados para editar ou modificar as referências”. Dentre os benefícios advindos com a utilização do *RefShare* podemos citar a facilidade na difusão da informação e disponibilização de condições para a criação de um ambiente de pesquisa colaborativo.

3 TECNOLOGIAS GRATUITAS UTILIZADAS NO DESENVOLVIMENTO DE SISTEMAS WEB

Neste capítulo são descritas algumas definições e características de várias tecnologias empregadas, atualmente, no desenvolvimento de sistemas para a web. Serão abordadas apenas tecnologias disponibilizadas gratuitamente, uma vez que as proprietárias não se coadunam com o escopo deste trabalho.

3.1 AMBIENTES DE DESENVOLVIMENTO INTEGRADO

Também conhecidos como IDE, do inglês *Interface Development Environment*, os ambientes de desenvolvimento integrado fazem parte de uma família de aplicativos voltados ao desenvolvimento de softwares. Sua finalidade precípua é tornar mais fácil, elegante, robusto e ágil o trabalho dos desenvolvedores, com significativa redução no tempo gasto no processo produtivo de softwares.

Como sugere sua designação, um bom ambiente de desenvolvimento integrado é formado por várias ferramentas que trabalham interligadas a fim de acelerar o processo produtivo e melhorar consideravelmente a qualidade do software produzido. As ferramentas básicas que compõem uma IDE são: um editor de textos (para o código-fonte), uma ferramenta de apoio à gestão de banco de dados, um compilador/interpretador, um depurador e um sistema de controle de versão (TONU, 2012).

3.1.1 NetBeans

O NetBeans é um ambiente de desenvolvimento integrado (IDE), open-source, escrito totalmente em java, atualmente encontra-se na versão 7.3. “É uma IDE madura, desenvolvida pela Sun, [...] e que vem se destacando cada vez mais por ser uma complexa, gratuita e ao mesmo tempo uma ferramenta de desenvolvimento multiplataforma”. (GONÇALVES, 2006).

O que faz o NetBeans atrair desenvolvedores de todo o mundo é que, além de competir com as melhores ferramentas de desenvolvimento java, disponibiliza diversas funcionalidades, tais como: um editor amigável de telas AWT/Swing; suporte completo ao Java Enterprise Edition; integração com banco de dados; plug-ins de diversos tipos que

Corporation e Microsoft Access. (DANTAS, 2011).

Oferece suporte para desenvolvimento de aplicativos voltados aos dispositivos móveis. Algumas razões que levam o usuário a desenvolver esses aplicativos é que existem mais de 2 (dois) bilhões de dispositivos a espera de uma *app* e o Netbeans oferece, gratuitamente, um conjunto de ferramentas para este tipo de desenvolvimento, muito fácil e poderoso, com analisador de código, ferramentas de depuração e vários outros componentes.

Segundo ensinamento de Böck (2012) a “base da plataforma NetBeans e sua arquitetura modular é chamado *NetBeans Runtime Container*”. A quantidade de módulos que sustentam esta complexa arquitetura são em número de 5 (cinco). A saber:

- *Bootstrap* - responsável por inicializar o sistema, é o primeiro módulo a ser executado, pois é ele que prepara o ambiente para a execução dos demais módulos.
- *Startup* - tem como finalidade a inicialização dos módulos e o sistema de arquivos.
- *Module System API* - esta interface faz o gerenciamento dos módulos (configurações e dependências).
- *File System API* - viabiliza a criação de um sistema de arquivos virtual a fim de permitir acesso ao Netbeans de forma independente.
- *Lookup & Utilities API* - favorece a comunicação entre os módulos e a pesquisa de forma independente da plataforma Netbeans.

No entendimento de Böck (2012), que é um admirador da plataforma em tela, é válido o uso desta, não importando se o aplicativo a ser desenvolvido seja muito pequeno ou muito grande, em virtude das possibilidades oferecidas pelo Netbeans, as quais disponibilizam soluções simples para os problemas e desafios encontrados no percurso do processo de desenvolvimento o que contribui para manutenção de um elevado grau de satisfação de seus usuários.

3.1.2 Eclipse

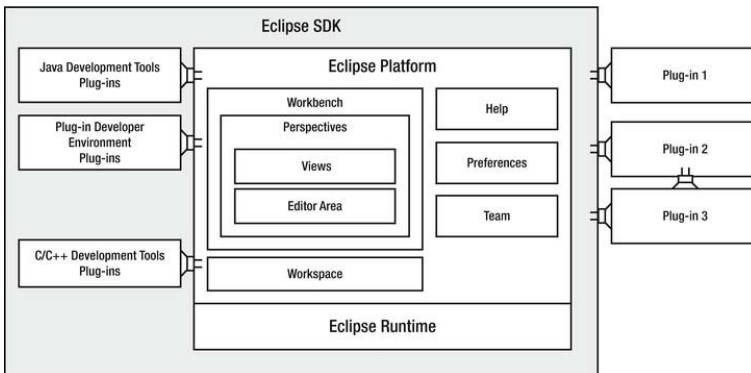
É uma IDE desenvolvida em Java, seguindo o modelo *open source* de desenvolvimento de software. A última versão lançada é a 4.3

(Eclipse Kepler). O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade. O gasto inicial da IBM no produto foi de mais de 40 milhões de dólares. (HOLZNER, 2004).

Atualmente conta com a Fundação Eclipse, criada em 2004, para gerenciar de forma independente a comunidade de indivíduos e organizações que se engajaram neste projeto de construção e manutenção de uma plataforma de desenvolvimento aberta, com estruturas extensíveis e ferramentas que auxiliam na construção, implantação e gerenciamento de software durante todo o seu ciclo de vida.

Um ambiente de desenvolvimento baseado em Eclipse tem como estrutura um conjunto de *plug-ins*, os quais são responsáveis por implementar as funcionalidades de que dispõe. Assim, o usuário poderá personalizá-lo de acordo com as suas necessidades, de modo simples e ágil. A atualização pode dar-se de modo automático, fazendo uso do gerenciador de atualização da plataforma, ou, então, usando instaladores nativos. A Figura 8 ilustra, de modo abrangente, a estruturação em comento.

Figura 8 – Visão geral da arquitetura da plataforma Eclipse



Fonte: Cinar (2012).

Acompanhando as tendências das tecnologias e a utilização destas pelos usuários, desenvolveu o projeto *Eclipse PHP Development Tools* (PDT), o qual teve excelente aceitação, uma vez que é um dos projetos mais baixados dos repositórios onde se encontra disponível. O Eclipse PDT é dotado, em sua distribuição padrão, das seguintes funcionalidades: suporte às linguagens de script PHP e Javascript,

suggerências para autocompletamento de código e depuração (local e remota) do código. (ECLIPSE FOUNDATION, 2013).

Uma exploração sucinta da IDE passa pelos seus intrínsecos conceitos e que são elucidados na sequência. *Workspace* é o diretório no sistema de arquivos local onde os projetos em desenvolvimento (código fonte e configurações do Eclipse) são armazenados. *Workbench* é o termo utilizado para referir-se à tela principal e esta é constituída por uma coleção de ferramentas, tais como: editores, menus, barra de ferramentas e outros. Editores são ferramentas utilizadas na elaboração dos arquivos de código-fonte e dos demais arquivos utilizados no projeto. Visualizações são opções de janela que permitem alterar (configurar) a vista do *workbench*, facilitando a navegação entre as várias possibilidades oferecidas pela IDE. Visualizações rápidas são vistas que são mostradas através de uma combinação de teclas e disponibilizam informações sobre o editor ativo ou sobre o elemento que se encontra sob o cursor. Apresenta uma variedade de menus para acesso às funcionalidades, além de barras de ferramentas para agilizar o acesso às mais comuns. Projeto é o termo que designa a organização e o agrupamento de arquivos que se relacionam para a consecução de um determinado objetivo. (CINAR, 2012).

Algumas das facilidades providas por este ambiente incluem funcionalidades para desenvolvimento de *Android APPs*, geradores e manipuladores de código, refatoração de código e atualização das dependências, entre outras que contribuem para o desenvolvimento mais rápido do código, permitindo que se dedique mais tempo à aplicação real.

3.1.3 Aptana Studio 3

O Aptana Studio 3, segundo Appcelerator (2012), é uma ferramenta de desenvolvimento profissional para a web, além de ser *open source*. Possibilita o desenvolvimento e teste da aplicação web completa usando um único ambiente. Oferece suporte para as mais recentes especificações de tecnologia dos navegadores, tais como: HTML5, CSS3, JavaScript, Ruby on Rails, PHP e Python.

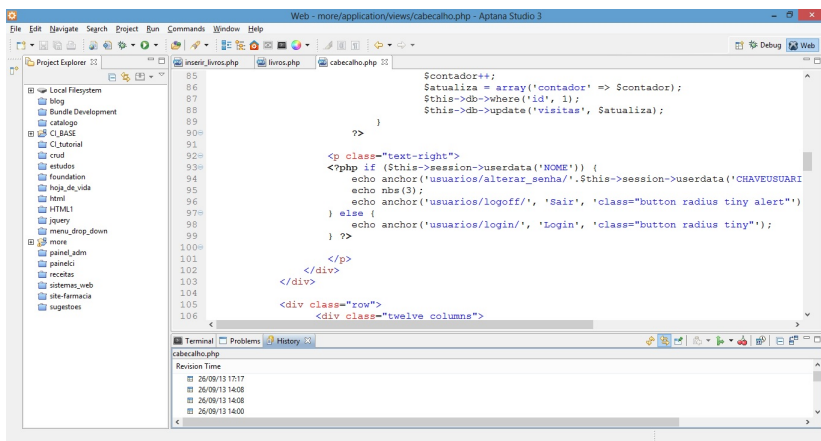
Auxilia na criação de HTML, CSS, JavaScript, PHP e Ruby. Ou seja, possui um serviço de *code completion* com centenas de comandos, mas sempre de acordo com a extensão do arquivo que está sendo editado. Suporta as mais recentes especificações do HTML5. Inclui informações sobre o nível de apoio para cada elemento nos principais

navegadores, informando ao usuário quais os navegadores mais populares que dispõem de suporte à execução do comando sugerido.

A definição de pontos de interrupção, para inspecionar variáveis e controlar a execução da aplicação, com um sistema integrado de Ruby & Rails e depuradores de JavaScript, facilita a depuração do código-fonte e a localização de eventuais falhas. Sua força é o javascript, uma vez que vem com um javascript de depuração integrado, fazendo do Aptana uma excelente escolha para o desenvolvimento em Ajax. Através do browser embutido, pode-se ver uma prévia do que projetar e testar sua funcionalidade. Após fazer a verificação do projeto em desenvolvimento, este pode ser sincronizado com o servidor web usando o FTP ou FTP seguro (SFTP).

Na sequência, a Figura 9 mostra a IDE em execução no sistema operacional *Windows 8*.

Figura 9 – Aptana Studio 3



Fonte: print screen da aplicação no sistema operacional *Windows 8*.

Sua interface gráfica é semelhante a da IDE Eclipse Java, uma vez que o Aptana Studio foi desenvolvido com base nesta plataforma, portanto, além de ser um software multiplataforma, pode ser instalado como um *plugin* do Eclipse ou então como uma versão autônoma. Preserva as principais qualidades do Eclipse e especializa-se em desenvolvimento de sistemas web.

Possui mais de 30 (trinta) temas para configurar a aparência de sua interface gráfica que podem ser alterados com apenas um clique do

mouse. Existe, ainda, a possibilidade de personalização de temas e seu armazenamento para utilização em outra instalação do Aptana.

Para Deuling (2013) este ambiente é provido de editores específicos para cada tipo de desenvolvimento suportado conforme a linguagem de programação utilizada. Desta forma, pode oferecer destaque no código-fonte e opção para expandir ou recolher os nós do arquivo em edição. Dentre esses editores podemos citar os utilizados para desenvolvimento HTML, CSS, PHP, JavaScript e Ruby on Rails.

A definição de projeto para o Aptana Studio está ligada à ideia de um conjunto de arquivos que contribuem para a construção de uma aplicação web. Ao criar um projeto, o usuário deverá informar sua natureza para que a IDE possa indexar os arquivos de forma coerente e apresentar as sugestões para autocompletamento do código em desenvolvimento. Esses projetos podem ser criados, excluídos, importados, exportados, abertos, fechados, depurados e refatorados.

Disponibiliza uma interface gráfica, chamada de *Studio Ajax Monitor*, onde o usuário poderá obter todas as informações sobre as requisições ajax que estão em andamento entre o agente de usuário e o servidor. Dispõe, também, de uma ferramenta para conexões FTP, para aqueles que utilizam servidores web de terceiros, simplificando o processo de atualização de arquivos, uma vez que, através da sincronização automática, é capaz de identificar os arquivos do projeto que sofreram algum tipo de modificação e efetuar a transferência somente destes, retirando do usuário a tarefa de identificá-los e selecioná-los manualmente, o que pode causar transtornos quando tratar-se de aplicações mais complexas.

A utilização do Aptana Studio 3 para o desenvolvimento de uma aplicação web utilizando PHP, segundo Deuling (2013), aumenta a satisfação do usuário, diante da gama de funcionalidades disponíveis e da facilidade em utilizá-las.

3.1.4 Astah Community

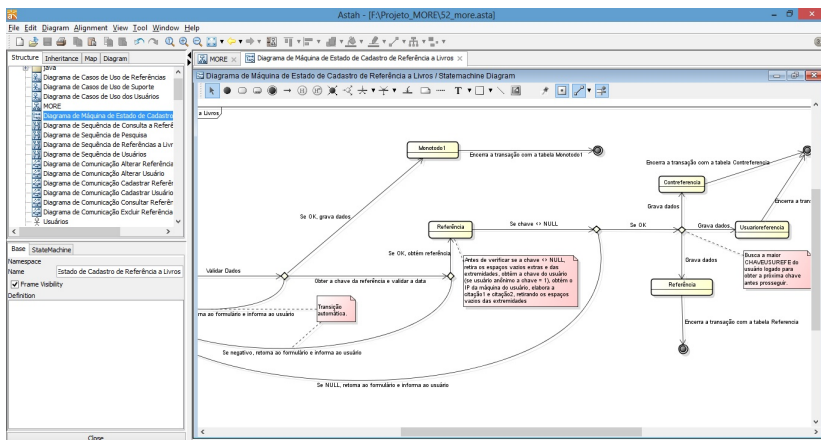
A ferramenta CASE de modelagem, Astah Community, é *open source* e é utilizada para o desenvolvimento da modelagem de software, utilizando-se da UML (*Unified Modeling Language*). Foi desenvolvido pela *Change Vision, Inc* que disponibiliza outras duas versões, porém estas são comerciais, o astah professional e o astah UML. É flexível e extensível e contém vários recursos. Esta versão do software permite apenas o desenvolvimento dos diagramas: diagrama de casos

de uso, diagrama de classe, diagrama de sequência, diagrama de estados, diagrama de atividades, diagrama de componentes, diagrama de implantação, diagrama de estrutura composta, diagrama de comunicação e diagrama de pacote (BRONDANI et al., 2011). Enquanto que a versão astah professional, além dos diagramas acima, permite também o desenvolvimento de fluxograma, diagrama de fluxo de dados (DFD), diagrama entidade-relacionamento, CRUD, mapa mental e tabela de requisitos (CHANGE VISION, 2009).

Ressalta-se que a UML é uma linguagem que faz uso de uma série de símbolos e regras padronizadas utilizadas na construção de módulos que compõem a estrutura de projetos de software independente de sua complexidade. Para Booch, Rumbaugh e Jacobson (2012), “é adequada para a modelagem de sistemas de informação corporativos a serem distribuídos a aplicações baseadas na web e até sistemas complexos embutidos de tempo real”.

A Figura 10 mostra uma visão do astah community em execução no sistema operacional *Windows 8*.

Figura 10 – Astah Community



Fonte: print screen da aplicação em execução no sistema operacional *Windows 8*.

A tela principal da IDE é dividida em 4 (quatro) subtelas. Na parte superior encontra-se a barra de tarefas, constituída de botões que servem de atalho para as principais funções. Na lateral superior esquerda é disponibilizada a estrutura, em camadas, do projeto em

edição. Na lateral inferior esquerda pode-se verificar e alterar as propriedades do diagrama, ou de algum de seus elementos, em construção no editor de diagramas, que ocupa a lateral direita da tela. É o editor padrão do astah e é nele que os diagramas e modelos são construídos.

Segundo ensinamentos de Change Vision (2012) o editor de diagramas disponibiliza uma barra de ferramentas personalizada para cada tipo de diagrama, contendo apenas os botões que representam os símbolos UML que poderão ser inseridos naquele determinado diagrama. A versão *community* do astah permite a exportação dos diagramas apenas como imagem (png ou jpeg) ou a impressão destes.

A utilização dos modelos e/ou diagramas na construção de um software é de fundamental importância para a sua qualidade, facilitando a compreensão da estrutura e do funcionamento, uma vez que eles representam, simplificada e, a realidade que se busca alcançar.

3.2 PHP

PHP (originalmente significava *Personal Home Page*, porém de acordo com a convenção para atribuição de nomes recursiva passou a significar *PHP Hypertext Preprocessor*) é uma linguagem de script *open source* de uso geral, muito utilizada e especialmente guardada para o desenvolvimento de aplicações Web embutida dentro do HTML. Pesquisas sobre servidores web realizadas por Netcraft (2013), desde 1995, mostram que o PHP em janeiro deste ano já estava sendo usado por 244 milhões de sites, o que significa que quase 40% dos servidores web estão utilizando esta linguagem de script no lado servidor.

Segundo Gilmore (2011) uma das vantagens do PHP é que se pode inserir o código PHP diretamente ao longo do HTML. Para que o código faça tudo, a página precisa ser transformada em ferramenta PHP para ser interpretada.

O que distingue o PHP de algo como Javascript no lado do cliente é que o código é executado no servidor, gerando HTML que é então enviado para o cliente. O cliente receberá os resultados da execução desse script, mas não saberia como é o código fonte. Pode-se inclusive configurar o servidor para processar todos os arquivos HTML como PHP, e então impossibilitará os usuários descobrirem se foi usada essa linguagem ou não.

A melhor coisa em usar PHP está no fato de ele ser extremamente simples para um iniciante, com uma curva de aprendizagem bem acentuada, mas oferece muitos recursos para o programador profissional.

Para Achour et al. (2013) o PHP é focado para ser uma linguagem de script do lado do servidor, portanto, pode fazer qualquer coisa que outro programa CGI (*Common Gateway Interface*) pode fazer, como: coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies. Mas o PHP pode fazer muito mais.

Esses são os maiores campos onde os scripts PHP podem ser utilizados:

- Script no lado do servidor (server-side). Este é o mais tradicional e principal campo de atuação do PHP. Necessita-se de três coisas para a realização do trabalho. O interpretador do PHP (como CGI ou módulo), um servidor web e um browser. Basta rodar o servidor web conectado a um serviço PHP instalado. É possível acessar os resultados do programa PHP com um browser, visualizando a página através do servidor web. Todos eles podem rodar em um servidor local, na própria máquina em que está sendo desenvolvido, em casa, para que se possa verificar o resultado do que está sendo desenvolvido com o PHP.
- Script de linha de comando. Pode-se fazer um script PHP funcionar sem um servidor web ou browser. A única coisa necessária é o interpretador. Esse tipo de uso é ideal para script executados usando o *cron* ou o Agendador de Tarefas (no Windows). Esses scripts podem ser usados também para rotinas de processamento de texto.
- Escrevendo aplicações desktop. O PHP provavelmente não é a melhor linguagem para criação de aplicações desktop com interfaces gráficas, mas para um bom conhecedor da linguagem que opte por usar alguns dos seus recursos avançados nas suas aplicações do lado do cliente, pode-se usar o PHP-GTK para escrever programas assim.

Conforme nos ensina Gilmore (2011) verifica-se que as características da linguagem tendem a cair em 4 (quatro) categorias chave: praticidade (a linguagem foi criada com a praticidade em mente, que permite ao usuário construir aplicações poderosas mesmo com um mínimo de conhecimento), poder (desenvolvedores PHP têm a seu dispor mais de duas centenas de bibliotecas que vai desde a criação de manipulação de arquivos PDF até uma ampla lista de soluções de processamento de cartão de crédito), possibilidade (um usuário PHP está

sempre cheio de escolhas oferecidas pela linguagem) e preço (é disponibilizado gratuitamente).

Por sua variedade de possibilidades e qualidades, como por exemplo utilizar-se tanto da programação orientada a objetos como da programação estruturada, é que tem ganhado muitos adeptos. A título de ilustração, a Tabela 1 mostra uma pequena lista dos principais sites que estão utilizando PHP, adaptada de MacIntyre (2010).

Tabela 1 – Principais sites que usam PHP.

Nome	Descrição
Facebook	Redes sociais
Flickr	Compartilhamento de fotografias
Wikipedia	Enciclopédia colaborativa on-line
SugarCRM	Ferramenta de gestão de relacionamento com cliente
Dotproject	Ferramenta de gerenciamento de projetos
Drupal	Modelo para construção de sites web
Interspire	Boletins informativos e marketing por e-mail

3.3 CODEIGNITER

O CodeIgniter é um *framework* (um conjunto de ferramentas) de desenvolvimento de aplicações para desenvolvedores de sítios web usando PHP. Seu objetivo é permitir o desenvolvimento de projetos muito mais rápido do que se tivesse que escrever o código do zero, provendo um vasto conjunto de bibliotecas para tarefas comuns, bem como uma interface simples e uma estrutura lógica para acesso a essas bibliotecas.

Segundo Gabardo (2012) “o CodeIgniter foi criado por Rick Ellis, fundador e CEO da companhia norte-americana EllisLab Inc., que desenvolve projetos para a web incluindo um gerenciador de conteúdo chamado Expression Engine, de onde se originaram algumas funcionalidades do CodeIgniter”.

É um software gratuito e o usuário tem permissão para usar, copiar, modificar e distribuir tanto o software quanto sua documentação.

Para EllisLab (2013), o CodeIgniter apresenta as seguintes características:

- é um framework leve, não exigindo grande poder de processamento;

- possui um desempenho excepcional;
- dispõe de uma ampla compatibilidade com contas de hospedagem padrão que executam uma variedade de versões do PHP e configurações;
- requer uma configuração mínima;
- não exige a utilização da linha de comando;
- não exige a aderência às regras de codificação restritivas;
- não exige o aprendizado de uma linguagem de templates (embora um analisador modelo é opcional, se optar por seu uso);
- evita complexidade, favorecendo soluções simples;
- possui uma documentação clara e completa.

Algumas das características acima são ratificadas por Upton (2007), uma vez que afirma ser o CodeIgniter uma ferramenta que ajuda os programadores PHP no desenvolvimento de um trabalho melhor, simplificando a leitura e a atualização dos scripts, reduzindo a quantidade de código escrito, estruturando o software em camadas (utiliza o padrão MVC), disciplinando o código e tornando-o mais robusto. Não aconselha sua utilização por programadores iniciantes em PHP e HTML ou se for desenvolver sites com características pouco dinâmicas e/ou pouco interativas.

Sua instalação é simples. Basta baixar o arquivo *CodeIgniter_x.x.x.zip* localizado na página <http://ellislab.com/codeigniter> e descompactá-lo diretamente no diretório onde são armazenados os aplicativos web de seu servidor local. Este pacote traz consigo um subdiretório que contém o guia do usuário. Os arquivos de código-fonte podem ser manipulados através de um editor de texto qualquer, porém é aconselhável a utilização de uma IDE que suporte o PHP para potencializar a produtividade pelas facilidades oferecidas por estes ambientes.

O significado semântico do padrão MVC é mantido quando este é utilizado com CodeIgniter. Ou seja, significa uma divisão do sistema web em 3 (três) camadas, onde todas podem se comunicar entre si, porém esta abordagem não deve ser adotada uma vez que distancia-se de seu objetivo principal que é restrição de acesso à base de dados. O “M” relaciona-se a modelo que é a camada mais baixa, situando-se entre o controlador e a base de dados e cuja principal finalidade é executar as consultas à BD, solicitadas pelo controlador. O “V” refere-se a visão (interface gráfica do usuário) que é a camada mais alta e

sua finalidade é prover a interatividade entre o usuário e o sistema, recebendo e enviando informações para o controlador. O “C” designa controlador que é camada responsável por receber, validar, processar e distribuir toda a informação que transita no sistema e faz a conexão segura entre o modelo e a visão.

3.4 JAVASCRIPT

A linguagem JavaScript nasceu da necessidade de prover interatividade à uma página eletrônica, uma vez que a HTML não dispõe de elementos capazes de disponibilizar tal funcionalidade. Uma parceria da Netscape com a Sun Microsystems possibilitou seu lançamento em 1995. Porém este período foi marcado pela disputa de mercado entre os fornecedores de navegador o que levou a Microsoft a criar o JScript em resposta à concorrente. Outra limitação da HTML é sua impossibilidade de manipulação e processamento dos dados colhidos junto aos usuários, porém este problema foi contornado com a criação de várias linguagens, executadas no lado do servidor, aptas a receber e processar tais dados, das quais pode-se citar Java, ASP e PHP, entre outras.

Para o lado do cliente, Silva (2010) ensina que

JavaScript é uma linguagem desenvolvida para rodar no lado cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível porque existe um interpretador JavaScript hospedado no navegador.

Ressalta-se, ainda, que existem interpretadores JavaScript para serem utilizados no lado do servidor, os quais podem ser embutidos em aplicações gerais, como é o caso do *Spider Monkey* e do *Rhino*, o que possibilita usar JavaScript no lado servidor, porém não é muito usual.

A JavaScript oferece várias possibilidades aos seus usuários, das quais pode-se destacar a manipulação do conteúdo e da apresentação da página, alteração do comportamento do navegador, interação com formulários e interação com outras linguagens dinâmicas. Foi, justamente, o uso indiscriminado dessas possibilidades que gerou uma certa difamação da linguagem. Porém, com a adoção de padrões para a web e o desenvolvimento de acordo com estes padrões fizeram com que esta prática nefasta fosse revista. Essa revisão conduziu a adoção de 2 (dois) princípios básicos para o emprego da linguagem. O princípio da melhoria progressiva e o princípio de JavaScript não obstrutivo.

Os padrões web trouxeram consigo, também, a ideia de desenvolvimento em camadas, onde cada tecnologia empregada tem função específica, devendo ser evitada o emprego de uma em detrimento de outra. Assim, o HTML é responsável pela camada de estruturação dos conteúdos na página, as folhas de estilo são responsáveis pela camada de apresentação e as linguagens de scripts são responsáveis pelo comportamento da página.

O padrão de desenvolvimento em camadas trouxe a separação dos códigos de acordo com cada camada, os quais são inseridos na página através de enlaces. A adoção desta prática oferece algumas vantagens que são elicitadas por Silva (2010): “elimina a necessidade de repetição de códigos em diferentes páginas; facilita o reaproveitamento de trechos de códigos em outros projetos; facilita a busca e correção de eventuais bugs nos códigos; facilita a manutenção e o entendimento dos códigos”.

Para adiconar JavaScript a uma página web existem três maneiras diferentes. A primeira, e mais recomendada, é através de arquivos próprios da linguagem, fazendo uma chamada específica na página que o utilizará. A segunda é inserindo o código na *tag* `<head>`, porém delimitado pela *tag* `<script>` e especificando que seu tipo é JavaScript. A terceira é *inline*, no corpo do documento e segue as recomendações da forma anterior.

Possui apenas três tipos primitivos de dados: *string*, *boolean* e *number*. Porém, para executar sua tarefa, a linguagem dispõe de funções, laços de repetição, condicionais e, inclusive, objetos. Uma vez que trata-se de uma linguagem fracamente tipada, então o que diferencia os tipos de dados contidos numa variável é o valor literal a ela atribuído e/ou o contexto em que é utilizada (POWERS, 2010).

Apesar de não tratar-se de uma linguagem orientada a objetos, dois dos seus tornaram-se muito populares, confundindo os mais desavisados que pensam tratar-se de tecnologia diferente: o AJAX e o JSON.

AJAX é uma forma abreviada para referir-se ao termo em inglês *Asynchronous JavaScript and XML*. É definido por Niederauer (2007) como “o uso sistemático de JavaScript [...] para tornar o navegador mais interativo com o usuário, utilizando-se solicitações assíncronas de informações”. Uma das principais características do Ajax é a possibilidade de trocar dados com o servidor, apresentando-os ao usuário (s.f.c.), sem a necessidade de carregar toda a página, apenas atualizando os elementos que sofreram alterações.

JSON é o termo que refere-se à expressão em inglês *JavaScript*

Object Notation, ou seja, Notação para Objetos em JavaScript. Para Silva (2012a) “trata-se de uma técnica JavaScript que usa sintaxe leve e simples para armazenar dados. Tal como XML, JSON está estruturado de modo a ser facilmente lido, interpretado e processado por máquinas”. Vale ressaltar que sua estrutura suporta uma coleção de pares nome/valor e *arrays*, mas compostos somente de valores literais JavaScript.

3.5 HTML

HTML é um acrônimo para *HyperText Markup Language*, cuja tradução para o português é Linguagem para Marcação de Hipertexto. No contexto deste trabalho, hipertexto pode ser entendido como todo o conteúdo inserido em um documento para a web e tem como principal característica a possibilidade de interligação a outros documentos lá existentes. Esta ligação é viabilizada através dos enlaces existentes nas páginas web.

A web foi inventada em 1992 por Sir Tim Beners-Lee que vislumbrou a possibilidade de interligação de hipertextos em computadores diferentes com a utilização de enlaces globais, conhecidos atualmente como hiperlinks. Desenvolveu um software próprio e um protocolo para recuperar hipertextos, denominado HTTP (*HyperText Transfer Protocol*) e o formato do texto que este protocolo aceitaria foi chamado de HTML.

Dois anos mais tarde, em 1994, foi criado o *World Wide Web Consortium* (W3C), um consórcio internacional que aglutina vários setores, com a finalidade de desenvolver a web a seu máximo potencial, através da criação de normas e especificações aplicáveis aos mais diversos setores e segmentos da web. Desde então a HTML passou por diversas atualizações até chegar ao que hoje se denomina HTML5.

Segundo Silva (2011) a especificação para a HTML5 está estruturada em 10 seções, a saber:

- *Infraestrutura comum*: define terminologia, classes, algoritmos, sintaxes e partes comuns das especificações;
- *Semântica, estrutura e APIs para documentos HTML*: definem as funcionalidades do DOM HTML e dos elementos HTML em geral;
- *Elementos HTML*: explicam o significado de cada um dos elementos HTML;

- *Microformatos*: a especificação para a HTML5 prevê um mecanismo para marcar informações sobre o documento, no formato nome/valor, para serem lidas por máquinas;
- *Carregamento de páginas web*: define as muitas funcionalidades relacionadas ao tratamento de páginas web pelos diferentes dispositivos;
- *APIs para aplicações web*: descrevem as funcionalidades básicas para desenvolvimento de scripts em aplicações HTML;
- *Interação com o usuário*: descreve os diferentes mecanismos de interação do usuário com um documento HTML;
- *APIs para comunicação*: tratam dos mecanismos de comunicação entre aplicações HTML executando em domínios diferentes e no mesmo cliente
- *Sintaxe HTML*: descreve a sintaxe HTML;
- *Sintaxe XHTML*: descreve a sintaxe XHTML.

O que a HTML5 busca, basicamente, em sua padronização é oferecer recursos gráficos, de vídeo, de áudio e de geolocalização para evitar a dependência, de desenvolvedores web, das tecnologias proprietárias (por exemplo, o Flash da Adobe System, o QuickTime da Apple e o Silverlight da Microsoft).

Segundo Vaughan-Nichols (2010) a “HTML5 vai apoiar as mais recentes tecnologias móveis, como os serviços de geolocalização e baseados em localização (LBS), bem como gráficos vetoriais escaláveis (SVG). SVG é um formato de arquivo baseado em XML aberto, produz gráficos compactos e de alta qualidade”.

O que tem deixado os desenvolvedores mais contentes é o fato da HTML5 estar implementando nativamente muitas das funcionalidades que, no passado recente, os desenvolvedores queriam ou precisavam codificar, porém esbarravam, muitas vezes, na variedade de navegadores que seus usuários utilizavam.

Uma de suas novas potencialidades que é exaltada por Anthes (2012) é o “recurso conhecido como armazenamento local. A HTML5 permite o armazenamento persistente de dados estruturados da sessão no cliente web. Ao contrário dos cookies, esta capacidade permite o armazenamento de grandes quantidades de dados para acesso mais tarde, quando uma conexão com um servidor Web pode não estar disponível ou não ser desejável”.

Atualmente, o W3C conta com um grupo permanente (WHATWG) para analisar as solicitações e propor novos padrões, de modo a adequar a HTML às tecnologias emergentes.

3.6 CASCADING STYLE SHEETS (CSS)

Cascading Style Sheets, cuja tradução é Folhas de Estilo em Cascata, também são definidas e padronizadas pelo W3C e têm por finalidade devolver à marcação HTML/XML o propósito inicial da linguagem. A HTML foi criada para ser uma linguagem exclusivamente de marcação e estruturação de conteúdos. Isso significa que, segundo seus idealizadores, não cabe à HTML fornecer informações ao agente do usuário sobre a apresentação dos elementos. Por exemplo: cores de fontes, tamanhos de textos, posicionamentos e todo o aspecto visual de um documento não devem ser funções da HTML. Cabem às CSS todas as funções de apresentação de um documento, e essa é sua finalidade maior. Daí a já consagrada frase que resume a dobradinha CSS + HTML: “HTML para estruturar e CSS para apresentar” (SILVA, 2012b).

Segundo Amaral (2009), CSS “é uma linguagem de estilo utilizada para descrever a apresentação de um documento escrito em uma linguagem de marcação, como HTML, XHTML ou XML. Essas definições são aplicadas a documentos de diversas maneiras, alterando a forma como as informações são apresentadas”, ou seja, é responsável pela formatação do design de uma página web.

Existem três formas de inclusão das CSS em um documento: estilos externos, cuja inclusão é realizada através do uso da *tag* `<link>` que associa às folhas de estilo em cascata contidas no documento referenciado ao documento que o referenciou; estilos incorporados no próprio documento através da *tag* `<style type=“text/css”>`; e, estilos *inline*, definindo atributos a qualquer elemento do documento HTML, porém esta forma deve ser utilizada com critério (não é aconselhável seu uso desta maneira).

Para Rebitte e Vinicius (2006) trata-se de “uma poderosa e precisa linguagem de formatação, que reduz drasticamente o consumo de banda do usuário pelo site”. É muito bem aceita pela maioria dos navegadores e contribui para a acessibilidade, uma vez que não altera a estrutura da informação disponibilizada na página web.

Especial atenção deve ser dispensada na utilização das CSS para formatar um documento, tendo em vista que pode ocorrer conflitos

e o navegador deve seguir determinadas regras para solucioná-los. É neste contexto que o termo cascata assume seu significado semântico, definindo os graus de importância de cada estilo. Assim o estilo do autor tem precedência sobre o estilo do leitor, e estes sobre o estilo padrão do navegador. No documento há o encadeamento de acordo com a sequência de inclusão, de modo que o último a ser incluído na página terá a maior precedência.

3.7 ZURB FOUNDATION

Zurb Foundation é um framework CSS que facilita sobremaneira a definição do design de uma página, além de engrossar as fileiras dos aplicativos *open source*.

Sua concepção está baseada em um layout de grade flexível de 12 colunas que pode ser definido para um tamanho arbitrário até que se alcance a largura máxima da linha, que também são facilmente aninhadas, possibilitando a construção de layouts complexos sem a criação de uma série de elementos personalizados.

O *Foundation* dispõe de dezenas de estilos para facilitar a criação rápida de protótipos de um site, vários tamanhos de botões e estilos, guias, elementos de formulário personalizados, caixas de diálogo modais, *sliders* de imagens e muito mais. Todos os elementos do protótipo estão com seus códigos prontos para produção, e são facilmente adaptáveis ao projeto em desenvolvimento.

Para atender também a demanda dos dispositivos móveis o *Foundation* foi construído a partir do zero para suportar qualquer tipo de dispositivo, qualquer tamanho de tela, com qualquer resolução, uma vez que usa a abordagem de percentuais em suas definições de valores de tamanhos. Assim pode-se iniciar a construção do sistema para uma gama de dispositivos sem a preocupação de adaptar-se a cada um deles, ou pode-se criar um site totalmente personalizado para um dispositivo específico. Estão incluídos, ainda, estilos CSS para ocultar e mostrar elementos em diferentes tipos de dispositivos e navegadores, de acordo com as possibilidades de cada um deles (ZURB, 2013).

3.8 SGBDR MYSQL

O Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR) MySQL “é um servidor de banco de dados relacional que

se aproxima cada vez mais das facilidades encontradas em produtos proprietários competitivos a cada versão” (GILMORE, 2011).

As características a seguir mencionadas ajudam a tornar o SGBD MySQL um dos mais populares. Além do preço (gratuito para vários usuários) podemos citar:

- *Flexibilidade*: independente do sistema operacional utilizado, é bem provável que o MySQL atenda as necessidades;
- *Capacidade*: apesar de o foco inicial ter sido o desempenho (e a velocidade de desempenho é mantida até hoje) no decorrer do tempo foram adicionadas várias capacidades que o habilitam competir com sistemas proprietários;
- *Recursos de SQL a nível de empresa*: oferece recursos avançados tais como *subqueries*, *views* e *stored procedures*;
- *Indexação e busca full-text*: este recurso melhora enormemente o desempenho de mineração de dados a partir de colunas baseadas em texto;
- *Query caching*: é uma das maiores melhorias em velocidade do MySQL. Simples e altamente efetiva quando habilitada, a *query caching* faz com que o MySQL armazene consultas *select*, juntamente com seus resultados correspondentes, em memória;
- *Replicação*: permite que um banco de dados localizado em um servidor MySQL seja duplicado em outro servidor MySQL, o que oferece um grande número de vantagens;
- *Segurança*: oferece uma quantidade enorme de opções de segurança e de configuração, permitindo que se tenha o controle total sobre praticamente todos os aspectos de sua operação.

Sua arquitetura está organizada em 3 (três) camadas. A camada superior dispõe dos serviços mais comuns de um SGBDR, tais como segurança e gerenciarneto de conexão. A segunda camada contém o diferencial do MySQL em relação aos demais, pois é nela que se encontram todas as funções internas, os códigos para análise de consulta, otimização de cache, além das *stored procedures*, *triggers* e *views*, entre outras funcionalidades. A terceira camada contém todos os mecanismos de armazenamento e é responsável pela gravação e recuperação dos dados (SCHWARTZ et al., 2008).

Para gerenciar as conexões e garantir a segurança, atribui a cada conexão cliente uma *thread* dedicada, que são mantidas em cache para

evitar sua criação e destruição a cada nova conexão. Para otimizar as consultas cria uma árvore de decisão, analisa e elabora o plano de execução da consulta. O controle de concorrência é realizado através de dois tipos de bloqueio: o bloqueio compartilhado e o bloqueio exclusivo (ou bloqueio de leitura e bloqueio de gravação); este controle bloqueia apenas os recursos requeridos e não todos os recursos.

O MySQL possibilita a execução de transações, que segundo Schwartz et al. (2008) “é um conjunto de consultas SQL que são tratadas atômicamente, como uma única unidade de trabalho”. Assim, a consulta é tratada como se fosse uma tarefa única e deverá apresentar as características conhecidas como ACID, que quer dizer:

- *Atomicidade*: ou executa todo o conjunto de consultas, ou não executa nada.
- *Consistência*: o banco de dados deve passar de um estado consistente para outro estado consistente ao término da transação, assim uma transação não cria e nem destrói nada por si só.
- *Isolamento*: uma transação é completamente transparente a outra transação até que esteja completamente concluída.
- *Durabilidade*: é a garantia de persistência dos dados alterados pela transação no banco de dados, mesmo que venha a ocorrer uma falha no sistema.

Possui um sistema sofisticado para identificar *deadlocks* e reverter a transação que tem o menor número de bloqueios de linhas exclusivas, uma vez que o impasse não pode ser solucionado sem reverter uma das transações, total ou parcialmente. Enfim, adapta-se aos mais diversos tipos de ambientes, desde os mais exigentes, como aplicações web, aos mais comuns, como acionar aplicações embarcadas, ou sistemas redundantes de alta disponibilidade, ou processamento de transações *on-line*, e muitas outras (SCHWARTZ et al., 2008).

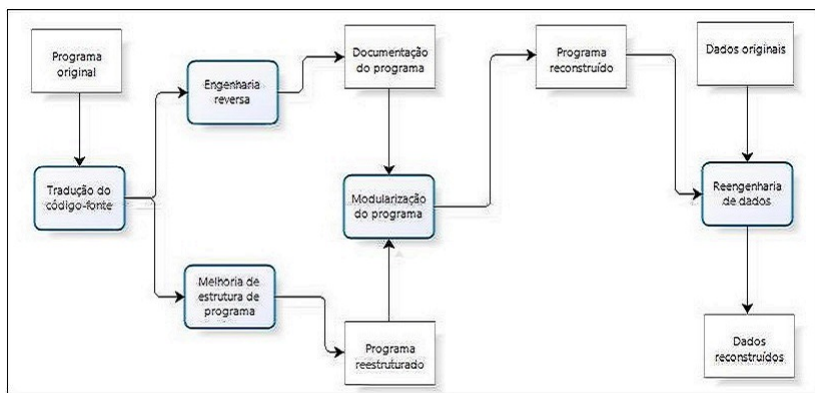
4 REENGENHARIA DO MORE

Neste capítulo será descrito todo o processo desenvolvido na reengenharia do MORE. Reengenharia de software, segundo Serra (2009), “é considerada uma modernização de software, ou seja, uma abordagem disciplinada para migrar softwares legados em softwares evolutivos”; e o entendimento de Sommerville (2011) é no sentido de aplicá-la para que se alcance uma “melhoria da estrutura e inteligibilidade” dos sistemas legados. Estes são sistemas que estão em produção há alguns anos e que, apesar de possuírem um alto valor de negócio, sua manutenção tornou-se muito complexa e dispendiosa.

Esta tem sido uma tendência em virtude das necessidades e anseios dos usuários, uma vez que as funcionalidades dos softwares devem adequar-se as tecnologias por eles utilizadas. Porém, recomenda-se uma avaliação precisa do contexto, uma vez que a reengenharia é recomendada somente para os casos em que o software possua um alto valor de negócio e uma baixa qualidade (manutenção difícil e de alto custo).

Da bibliografia pesquisada, verifica-se que o modelo de processo de reengenharia mais adequado ao escopo deste trabalho é o sugerido por Sommerville (2011), cuja representação pode ser verificada através da Figura 11, uma vez que o processo descrito por Pressman (2011) ou por Serra (2009) inicia-se por *Reestruturação de Documentos*, os quais não se encontram disponíveis no caso do MORE.

Figura 11 – O processo de reengenharia.



Fonte: Adaptada de Sommerville (2011).

Antes do detalhamento de cada fase componente do processo de reengenharia, salienta-se que o MORE, no início deste projeto encontrava-se em execução no servidor do RexLab/UFSC, em Florianópolis/SC, e que foi desenvolvido utilizando-se ferramentas e tecnologias gratuitas. O sistema teve sua construção apoiada na IDE Netbeans, fazendo uso, no lado servidor, da linguagem *Java Server Pages* (JSP) e, no lado cliente, JavaScript para adicionar dinamicidade às páginas web. A conformação das interfaces apresentadas aos usuários é baseada em folhas de estilo em cascata (CSS) e a estrutura das informações em HTML. O SGBDR MySQL foi utilizado para a construção e o gerenciamento de sua base de dados. Todas essas tecnologias permitem o desenvolvimento de um software conforme os padrões recomendados pelo W3C.

4.1 TRADUÇÃO DO CÓDIGO-FONTE

Esta etapa normalmente é realizada de forma automática por uma ferramenta de tradução. Nesta oportunidade o código-fonte é convertido para uma versão mais recente da linguagem de programação ou, então, convertido para uma linguagem mais atual (SOMMERVILLE, 2011). No entanto, esta tarefa foi efetuada manualmente, visto que apenas o arquivo compilado carregado no servidor encontrava-se disponível no início do processo.

Trata-se do arquivo “more.war” que foi descompactado e dele extraído os arquivos que o compõe, porém os arquivos “.java” já estavam compilados e assim somente encontravam-se disponíveis os arquivos “.class” que passaram por um tratamento através do aplicativo “jd-gui” a fim de converter estes arquivos “.class” em arquivos “.java”.

Infelizmente este processo não possibilitou a reversão do arquivo “more.war” a um estado consistente, uma vez que a compilação deste foi realizada sem a inclusão do código-fonte, ou seja, não foi possível reconstruir um projeto que pudesse ser compilado novamente pelo Netbeans.

Inicialmente foram identificadas todas as bibliotecas nativas da linguagem que foram utilizadas para que pudesse haver uma separação entre estas e as classes implementadas pelos desenvolvedores do MORE, cujas funcionalidades foram levantadas numa etapa posterior.

Na sequência, foi realizada a extração de todo código “HTML” e das folhas de estilo em cascata a fim de possibilitar a reconstrução das interfaces. Registra-se a utilização de dois arquivos JavaScript para

funções específicas. Um deles implementa as rotinas relativas ao envio de informações para serem processadas pelo *Google Analytics* e o outro as funções de suporte às funcionalidades do sistema.

4.2 ENGENHARIA REVERSA DO MORE

A engenharia reversa, segundo Serra (2009), “é um processo retroativo do ciclo de vida, que parte de um baixo nível de abstração (código-fonte ou executável) para um alto nível de abstração (modelos de projeto e especificação)”.

Uma análise criteriosa do código-fonte traduzido, conforme descrito na seção anterior, possibilitou elicitar os requisitos do sistema em produção e iniciar a elaboração da documentação. Nesta ocasião foi elaborada a modelagem do sistema, apoiada pela IDE *Astah Community*, e o documento de requisitos do MORE que engloba a modelagem em tela.

4.2.1 Documento de Requisitos do MORE

O documento de requisitos está estruturado da seguinte maneira:

- *Concepção do Sistema*: contém, basicamente, os objetivos do documento e convenções adotadas.
- *Visão Geral do Produto*: apresenta uma perspectiva do produto, além da descrição dos usuários e das funções do sistema.
- *Premissas, Restrições e Dependências*: essas estarão sendo adotadas durante a descrição dos casos de uso.
- *Arquitetura do Sistema*: apresenta a arquitetura do sistema.
- *Requisitos Funcionais*: apresenta todos os requisitos funcionais, os quais possuem o identificador [RFxxx], onde xxx são dígitos que variam entre 0 e 9.
- *Requisitos Não Funcionais*: apresenta todos os requisitos não funcionais, os quais possuem o identificador [RNFxxx], onde xxx são dígitos que variam entre 0 e 9.
- *Modelagem dos Casos de Uso*: apresenta os casos de uso do sistema com suas respectivas descrições.

- *Modelagem Lógica do Banco de Dados*: apresenta o modelo lógico do banco de dados do MORE.
- *Diagrama de Sequência e Comunicação*: apresenta os diagramas de sequência e comunicação do sistema.
- *Diagrama de Estado*: apresenta o diagrama de estado de cadastro de referência a livros.

4.2.1.1 Concepção do Sistema

Nesta seção do documento é especificado seu propósito de descrever os requisitos do sistema a fim de fornecer às partes interessadas uma visão das características do software a ser desenvolvido e, especificamente, aos desenvolvedores, as informações necessárias para a execução de seu projeto e implementação, bem como para a realização dos testes e homologação do produto do projeto.

Define seu público alvo, que são os clientes, usuários finais, patrocinadores e desenvolvedores a fim de mantê-los perfeitamente informados sobre as possibilidades que o MORE deve fornecer aos seus usuários.

Especifica seu escopo, o qual visa a elicitação de requisitos do MORE, cujo objetivo é dar maior confiabilidade e agilidade na elaboração de referências bibliográficas. Deverá implementar uma interface com alto grau de usabilidade e produzir automaticamente citações no texto e referências no formato ABNT para os tipos de fontes providas pelo sistema legado. Determina, ainda, que o MORE deverá ser implementado e validado de acordo com as normas do W3C e fazer uso das técnicas de elaboração de páginas web dinâmicas.

Traz uma tabela de definições, acrônimos e abreviações, bem como uma relação das referências bibliográficas utilizadas no suporte à sua elaboração.

4.2.1.2 Visão Geral do Produto

A perspectiva que se busca com a reengenharia do sistema atual é a adequação do código-fonte às tendências tecnológicas atuais, permitindo estruturar o código de modo a facilitar tanto a documentação quanto a manutenção, bem como flexibilizar a implementação de novas funcionalidades que vierem a ser necessárias, além de maior estabili-

dade do sistema. O resultado será a adequação da estrutura do código-fonte e da arquitetura do modelo funcional dos dados para que ambos permitam futuras alterações sem a necessidade de reescrita de todo o sistema.

Assim, pretende-se disponibilizar uma nova versão do MORE ao público em geral, para que possam desfrutar de uma ferramenta gratuita que os auxiliem na elaboração de referências bibliográficas, segundo as normas da ABNT, elaborada com tecnologias voltadas à web dinâmica.

4.2.1.3 Premissas, Restrições e Dependências

A linguagem de programação utilizada no lado servidor será o PHP e no lado cliente será o JavaScript. Portanto, para que qualquer usuário faça uso do sistema, necessita apenas de um navegador web com suporte a JavaScript.

Para que se mantenha o armazenamento dos dados, o servidor deverá suportar, além do PHP, um servidor web e o SGBDR MySQL 5.5.

As máquinas clientes deverão estar conectadas ao servidor web através da internet.

4.2.1.4 Arquitetura do Sistema

O sistema consiste em uma aplicação web que permitirá o acesso dos usuários através de uma interface simples e intuitiva. Na página inicial, além do menu e do contador de visitas, oferece um link para efetuar o login.

A arquitetura segue os moldes de uma aplicação web de três camadas, cujos dados serão armazenados na base de dados do sistema existente no servidor. Todas as informações necessárias para o funcionamento do sistema serão armazenadas nessa base de dados.

4.2.1.5 Requisitos Funcionais

Requisitos funcionais são as especificações das funcionalidades que o sistema deverá prover e seu comportamento frente a situações específicas. O documento de requisitos do MORE especifica cinquenta e oito requisitos funcionais, distribuídos em três módulos: suporte, con-

trole de usuários e controle de referências.

As Tabelas 2, 3 e 4 relacionam os requisitos funcionais elicitados de acordo com cada módulo.

Tabela 2 – Requisitos Funcionais do Módulo Suporte

Identificador	Descrição
[RF001]	Ajuda
[RF002]	Equipe
[RF003]	Índice
[RF004]	Contador de Visitas
[RF005]	Sobre o MORE
[RF006]	Links Úteis
[RF007]	Contato

Tabela 3 – Requisitos Funcionais do Módulo Controle de Usuários

Identificador	Descrição
[RF008]	Login
[RF009]	Logout
[RF010]	Cadastrar Usuário
[RF011]	Alterar Usuário
[RF012]	Consultar Referências
[RF013]	Pesquisa
[RF056]	Cadastrar Coleção
[RF057]	Alterar Coleção
[RF058]	Excluir Coleção

Tabela 4 – Requisitos Funcionais do Módulo Controle de Referências

Identificador	Descrição
[RF014]	Cadastrar Referência Livros
[RF015]	Alterar Referência Livros
[RF016]	Excluir Referência Livros
[RF017]	Cadastrar Referência Dicionário/Enciclopédia
[RF018]	Alterar Referência Dicionário/Enciclopédia
[RF019]	Excluir Referência Dicionário/Enciclopédia
[RF020]	Cadastrar Referência Relatório Técnico
[RF021]	Alterar Referência Relatório Técnico

Identificador	Descrição
[RF022]	Excluir Referência Relatório Técnico
[RF023]	Cadastrar Referência Teses e Dissertações
[RF024]	Alterar Referência Teses e Dissertações
[RF025]	Excluir Referência Teses e Dissertações
[RF026]	Cadastrar Referência Anais/Proceedings
[RF027]	Alterar Referência Anais/Proceedings
[RF028]	Excluir Referência Anais/Proceedings
[RF029]	Cadastrar Referência Capítulos de Livros
[RF030]	Alterar Referência Capítulos de Livros
[RF031]	Excluir Referência Capítulos de Livros
[RF032]	Cadastrar Ref. Verbetes Dicionários/Enciclopédias
[RF033]	Alterar Ref. Verbetes Dicionários/Enciclopédias
[RF034]	Excluir Ref. Verbetes Dicionários/Enciclopédias
[RF035]	Cadastrar Ref. Trabalhos Apres. em Congressos
[RF036]	Alterar Ref. Trabalhos Apres. em Congressos
[RF037]	Excluir Ref. Trabalhos Apres. em Congressos
[RF038]	Cadastrar Referência Revistas
[RF039]	Alterar Referência Revistas
[RF040]	Excluir Referência Revistas
[RF041]	Cadastrar Referência Jornais
[RF042]	Alterar Referência Jornais
[RF043]	Excluir Referência Jornais
[RF044]	Cadastrar Referência Artigo de Revistas
[RF045]	Alterar Referência Artigo de Revistas
[RF046]	Excluir Referência Artigo de Revistas
[RF047]	Cadastrar Referência Artigo de Jornais
[RF048]	Alterar Referência Artigo de Jornais
[RF049]	Excluir Referência Artigo de Jornais
[RF050]	Cadastrar Referência Homepages
[RF051]	Alterar Referência Homepages
[RF052]	Excluir Referência Homepages
[RF053]	Cadastrar Referência E-mail
[RF054]	Alterar Referência E-mail
[RF055]	Excluir Referência E-mail

4.2.1.6 Requisitos Não Funcionais

Requisitos não funcionais, segundo Sommerville (2011), “são restrições aos serviços ou funções oferecidos pelo sistema. Incluem res-

trições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas”. São especificados no documento catorze requisitos não funcionais referentes aos requisitos de processo, requisitos de produto (performance, segurança, usabilidade, distribuição e portabilidade) e requisitos externos, a seguir relacionados:

- [RNF001] *Linguagem de Programação*: o sistema será desenvolvido utilizando a linguagem de programação PHP, no lado servidor, e JavaScript, no lado cliente. A HTML e as CSS serão utilizadas para a construção das páginas web. (Prioridade: Essencial).
- [RNF002] *Modelagem*: o sistema deverá ser modelado utilizando a linguagem UML. (Prioridade: Importante).
- [RNF003] *Rápido tempo de resposta*: o sistema deverá realizar suas operações de maneira eficiente. Para que o usuário possa realizar suas tarefas rapidamente (tempo de resposta não superior a 15 segundos para uma conexão estável). (Prioridade: Desejável).
- [RNF004] *Acessos Simultâneos*: o sistema deverá suportar 30 (trinta) acessos simultâneos. (Prioridade: Importante).
- [RNF005] *Disponibilidade*: o sistema deverá estar disponível durante todo tempo (24 X 7). (Prioridade: Desejável).
- [RNF006] *Confidencialidade*: os usuários cadastrados no sistema deverão possuir um login e senha de acesso ao sistema para que assim possam ter acesso a todas funcionalidades do mesmo. (Prioridade: Essencial).
- [RNF007] *Integridade*: os dados armazenados e consultados deverão estar corretos em relação aos dados fornecidos ao sistema. (Prioridade: Essencial).
- [RNF008] *Confirmação de exclusão*: o sistema deverá alertar o usuário quando for solicitada a exclusão de uma referência, para que a exclusão seja feita o usuário deve confirmá-la na janela de alerta. (Prioridade: Importante).
- [RNF009] *Backup*: o sistema deverá realizar cópia de segurança dos dados no período compreendido entre 01:00 h e 05:00 h da manhã. (Prioridade: Importante).

- [RNF010] *Mensagens de Erro*: as mensagens de erro do sistema deverão ser precisas e construtivas, fazendo com que o usuário identifique sua origem e como proceder após sua ocorrência. (Prioridade: Desejável).
- [RNF011] *Interface do Usuário*: o sistema deverá prover uma interface para o usuário que seja simples, objetiva e intuitiva. (Prioridade: Desejável).
- [RNF012] *Implantação*: o sistema será posto em produção em um servidor web desta Instituição Superior de Ensino (IES) e deverá ser possível a sua utilização através dos navegadores web mais utilizados pelo público em geral. (Prioridade: Importante).
- [RNF013] *Tempo de Desenvolvimento*: o sistema deverá ser desenvolvido dentro do prazo limite de 8 meses. (Prioridade: Importante).
- [RNF014] *Veracidade dos Dados*: os dados inseridos no sistema serão um retrato fiel dos dados fornecidos pelos usuários, os quais serão os responsáveis por sua veracidade. (Prioridade: Essencial).

4.2.1.7 Modelagem dos Casos de Uso

O documento de requisitos especifica um caso de uso para cada requisito funcional do sistema, os quais foram dispostos em três diagramas de casos de uso, sendo um para cada módulo do sistema. Para cada caso de uso foi elaborada uma tabela com a sua respectiva descrição.

Os diagramas de casos de uso, no entendimento de Booch, Rumbaugh e Jacobson (2012), “são importantes para visualizar, especificar e documentar o comportamento de um elemento”.

As Figuras 12, 13 e 14 mostram os diagramas de casos de uso para os módulos do sistema.

A Tabela 5 mostra um exemplo de descrição de caso de uso, nesta oportunidade o UC014 (Cadastrar Referência a Livros), tendo em vista que o cadastro das demais referências seguem este padrão, que em tese é o mais complexo deste sistema.

Tabela 5 – Exemplo de Descrição de Caso de Uso

Campo	Descrição
Nome	Cadastrar Referência Livros
Identificador	UC014
Descrição	O usuário cadastra uma referência a livros.
Ator Primário	Usuários
Ator Secundário	–
Pré-condição	<ol style="list-style-type: none"> 1. Para que possa armazenar sua referência o usuário deverá estar logado; 2. São campos obrigatórios: Autor, tipo de autor, local (cidade), título, ano (aaaa) e editora; 3. São campo opcionais: 2º autor, 3º autor, país/estado/município, subordinação do autor, nr da edição, nr de páginas, nr de volumes, on-line, CD-ROM, URL, data, subtítulo, série e notas.
Fluxo Principal	<ol style="list-style-type: none"> 1. O usuário escolhe a opção Monografia no Todo → Livros; 2. O usuário informa as entradas necessárias; 3. O sistema valida as entradas; 4. A referência é armazenada e mostrada.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. Se o sistema verificar alguma inconsistência nas entradas, informa ao usuário e volta a mostrar o formulário; 2. O usuário corrige as inconsistências ou desiste pressionando o botão voltar.
Fluxo de Exceção	(ação inválida)
Pós-condição	Referência armazenada, mostrada ao usuário e, se este estiver logado, vinculada a ele.

4.2.1.8 Modelagem Lógica do Banco de Dados

Este modelo mostra como os dados serão armazenados em cada entidade da base de dados e quais são as relações existentes entre estas entidades.

O modelo conceitual/lógico enquadra-se em um nível médio de abstração e descreve quais objetos, propriedades e interrelacionamentos estão armazenados no banco de dados.

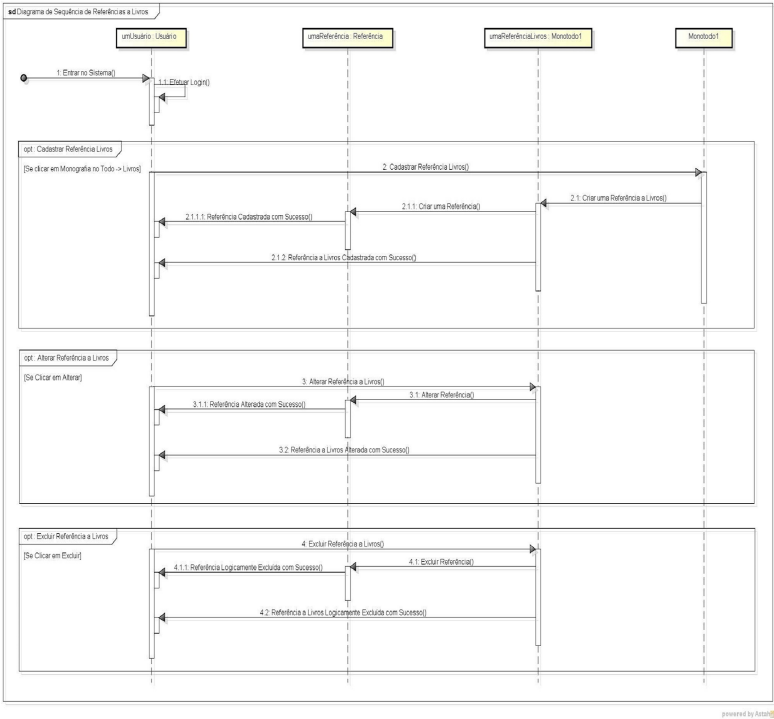
4.2.1.9 Diagramas de Sequência e Comunicação

Os diagramas de sequência e de comunicação fazem parte do portfólio de diagramas comportamentais da UML. Segundo Booch, Rumbaugh e Jacobson (2012), diagrama de sequência “ênfatisa a ordem temporal das mensagens”, enquanto que diagrama de comunicação “ênfatisa a organização estrutural de objetos que enviam e recebem mensagens”.

No documento de requisitos em tela encontram-se quatro diagramas de sequência e seis diagramas de comunicação que representam de forma gráfica como ocorre a parte do processo que cada um representa.

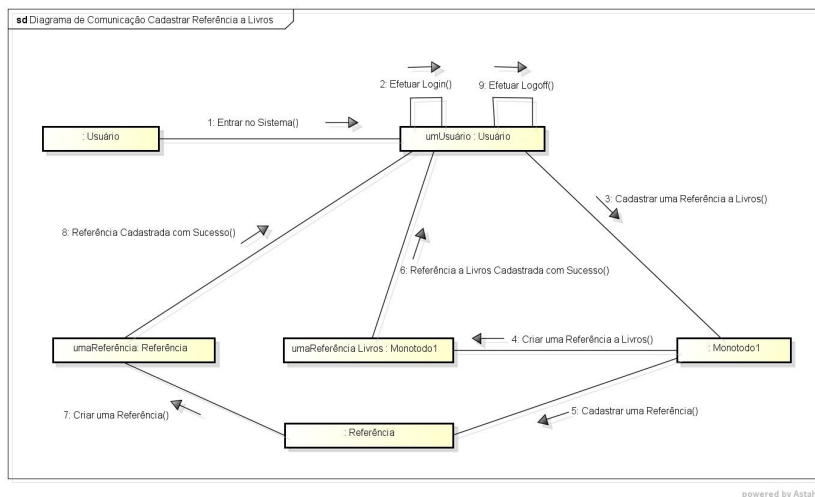
As Figuras 16 e 17 mostram, a título de exemplo, um diagrama de sequência e um diagrama de comunicação, respectivamente.

Figura 16 – Diagrama de Sequência de Referências a Livros



Fonte: Elaborada pelo autor.

Figura 17 – Diagrama de Comunicação Cadastrar Referência a Livros



Fonte: Elaborada pelo autor.

4.2.1.10 Diagrama de Estado

O diagrama de estado é, também, um dos diagramas comportamentais que a UML disponibiliza para representar comportamentos dinâmicos do sistema.

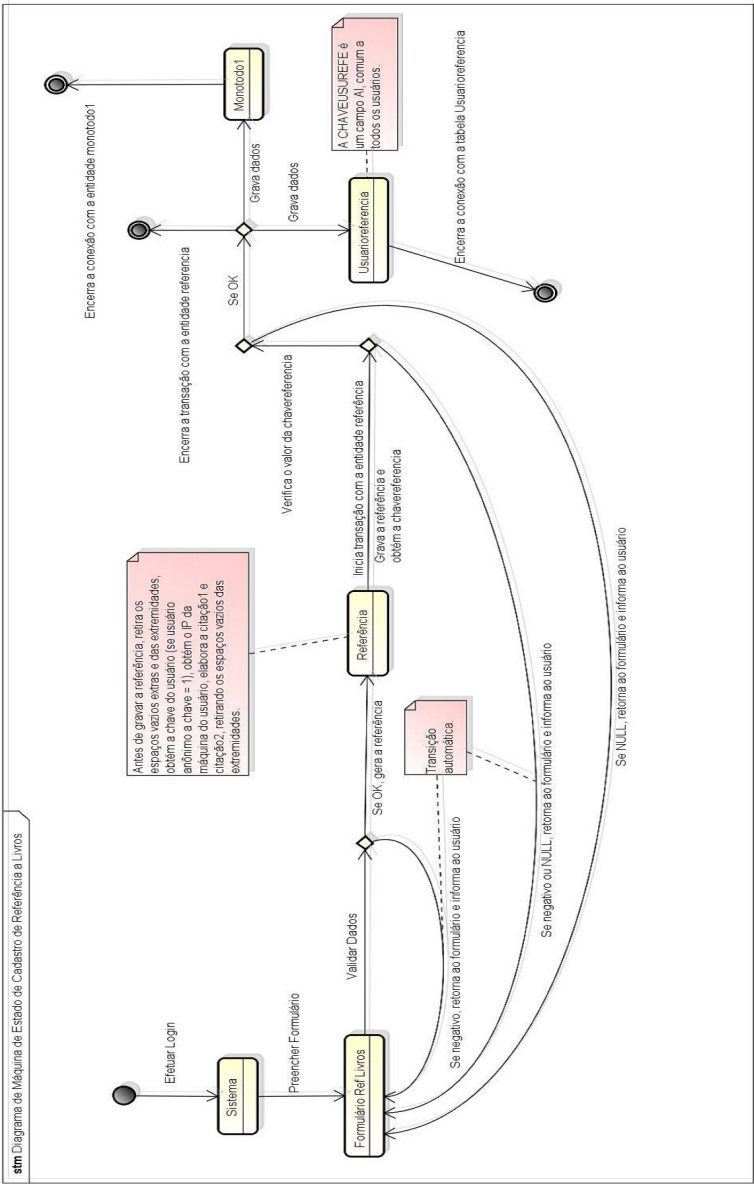
Para Booch, Rumbaugh e Jacobson (2012) um diagrama de estados “mostra uma máquina de estados” e “são úteis para a modelagem do tempo de vida de um objeto”, bem como, possibilita a visualização do “fluxo de controle de um estado para outro”.

Pode-se entender estado como uma situação durante o ciclo de vida do objeto, na qual ele espera que um evento ocorra, ou executa uma tarefa em resposta a um evento, ou ainda, satisfaz a uma determinada condição.

A passagem de um estado a outro é conhecida como transição, indicando que ações específicas, partindo do estado anterior, foram realizadas em resposta a um determinado evento, ou um conjuntos deles, e que as condições para que se alcance o próximo estado foram satisfeitas.

A Figura 18 exemplifica um diagrama de estado encontrado no documento de requisitos.

Figura 18 – Diagrama de Máquina de Estado de Cadastro de Referência a Livros



Fonte: Elaborada pelo autor.

4.3 MELHORIA DE ESTRUTURA DE PROGRAMA

A melhoria da estrutura do MORE deu-se basicamente em função da utilização do *framework CodeIgniter* que provê a estruturação do sistema em três camadas distintas e facilmente identificáveis. Para Sommerville (2011) a “estrutura de controle do programa é analisada e modificada para que se torne mais fácil de ler e entender”.

O *framework* abordado na Seção 3.3 utiliza-se da linguagem abordada na Seção 3.2 e estrutura o sistema em desenvolvimento segundo o *design pattern*, ou padrão de design, conhecido como MVC (*Model, View, Controller*). Este padrão foi desenvolvido pela Xerox PARC no período entre 1978 e 1979. Atualmente, vem sendo utilizado como solução para uma diversidade problemas.

O padrão de design MVC, segundo Gabardo (2012), “é uma solução para desacoplar as camadas de lógica e regras de negócio da camada de apresentação. Implementa três camadas distintas, sendo elas *Model, View* e *Controller*, cada qual com suas características e atribuições”.

4.3.1 Model

Os *Models* ou modelos são as classes responsáveis por toda conexão com a base de dados, organizando e simplificando esta tarefa complexa, além de reforçar a segurança, uma vez que implementa mais um filtro entre o usuário e a base de dados do sistema, sem comprometer o tempo de resposta.

Pode-se afirmar que são os elementos que compõem a camada de abstração de dados. Suas principais tarefas estão voltadas para a gravação e a recuperação dos dados. Apesar de estas tarefas poderem ser realizadas de outras maneiras, recomenda-se a utilização dos modelos, sempre que possível.

São eles que implementam a gravação das referências e dos dados dos usuários, bem como provê ao controlador os dados recuperados.

4.3.2 View

As *Views* ou visões fazem parte da camada de apresentação e são responsáveis pela interação do sistema com o usuário. Elas não devem comunicar-se diretamente com os *models* ou com a base de da-

dos, tampouco enviar dados a estes. Há uma certa flexibilidade na sua implementação com *CodeIgniter*, visto que podem utilizar HTML puro embebido com blocos de código PHP, além de *templates parser* ou *templates engines*.

São elas que apresentam ao usuário os formulários ou telas de informações, bem como realizam a coleta dos dados fornecidos pelos usuários e os remetem ao controlador específico.

4.3.3 Controller

Os *Controllers* ou controladores são as classes responsáveis pela comunicação entre as camadas do sistema. Implementam toda a lógica e as regras de negócio. É onde os dados são filtrados e/ou validados (se a validação não ocorreu no lado cliente). No caso do MORE, recebem os dados dos formulários, realizam o tratamento destes dados submetendo-os às lógicas e regras neles implementadas, elaboram (montam) as referências e as apresentam ao usuário através das *views*, bem como as submete ao respectivo modelo para gravação (referência e dados tratados).

Quando tratar-se de pesquisa o fluxo é inverso. Receberá o(s) termo(s) da pesquisa e fará uma consulta ao banco de dados através do modelo respectivo, o qual retornará os dados solicitados ao controlador, que os processará e os apresentará aos usuários através das *views*.

4.4 MODULARIZAÇÃO DE PROGRAMA

Modularização de programa é um processo manual em que partes convergentes e/ou afins do sistema são agrupadas de modo a eliminar as redundâncias, quando esta eliminação for uma ação recomendada (SOMMERVILLE, 2011).

A Subseção 4.2.1.5 traz um proposta de modularização do MORE, a qual foi adotada com o intuito de reunir funcionalidades afins num bloco mais amplo. A seguir elucida-se a abrangência de cada módulo.

No módulo suporte estão reunidas as funcionalidades gerais. Ali encontramos informações sobre o MORE, ajuda on-line, enlaces para páginas voltadas à referências bibliográficas, bem como um formulário de contato para acompanhar os anseios e inquietudes dos usuários.

O módulo controle de usuários agrupa as funcionalidades relativas aos dados individualizados. Este módulo é responsável por dispo-

nibilizar as possibilidades de cadastro de um usuário no sistema, sua autenticação e início da sessão, encerramento da sessão (por solicitação do usuário ou de forma automática ao sair da página), pesquisa de alguma referência persistida na base de dados e gerenciamento de suas referências (se estiver autenticado).

O módulo controle de referências possibilita e viabiliza a inserção, gravação e geração de referências cujas fontes estejam implementadas no sistema (nesta fase existem quinze tipos de referências implementadas).

4.5 REENGENHARIA DE DADOS

As alterações efetuadas na estrutura dos dados para adequar-se a nova estrutura do sistema foram poucas, o que não exigiu, efetivamente, uma reengenharia de dados.

As alterações que se fizeram necessárias foram:

- Criação de uma entidade para armazenar o contador de visitas;
- Criação de uma entidade para armazenar as coleções de usuários;
- Adição do campo *CHAVEPROJETO* à entidade *usuarioreferencia*;
- Adição do campo *NUMFOL* às entidades *monotodo1* e *monotodo3*.

5 RECONSTRUÇÃO E AMPLIAÇÃO DO SISTEMA

Este capítulo abordará a sequência de tarefas utilizadas para implementação, testes e entregas, que viabilizaram a reconstrução do MORE e a ampliação do seu portfólio de recursos. Para que houvesse uma entrega, esta passava, anteriormente, pelas fases de implementação e testes, inclusive testes de integração.

5.1 RECONSTRUÇÃO DO MORE

A reconstrução do sistema deu-se de forma iterativa e incremental seguindo os preceitos de metodologias ágeis, mais precisamente o *scrum* que é calcado em *sprints*, ou seja, durante o ciclo de desenvolvimento faz-se a previsão de entregas que requerem a inclusão de novas funcionalidades devidamente testadas.

Todas as páginas do sistema dispõem de um *breadcrumb* cuja finalidade é manter o usuário informado de sua localização na estrutura deste e oferecer um link direto para os níveis superiores desta estrutura.

Salienta-se que no canto superior direito encontra-se o nome do usuário, um enlace para alterar senha e um enlace para fechar a sessão, se estiver autenticado. Caso contrário, o nome será *Anônimo* e apenas o enlace de login será apresentado.

5.1.1 Módulo de Suporte

A primeira entrega constou da implementação do módulo de suporte onde, além das funcionalidades anteriores, foram acrescentadas as previstas relativas a repaginação visual do MORE, porém buscando manter suas características visuais.

Para levar a efeito a repaginação do MORE fez-se uso da tecnologia abordada na Seção 3.7 e algumas customizações para que a identidade com o design anterior pudesse ser mantida. As classes implementadas pelo *framework Zurb Foundation* fazem uso das CSS3, as quais adaptam-se muito bem à HTML5, que é a linguagem de marcação utilizada na estruturação das informações contidas nas páginas do sistema.

A Figura 19 mostra a tela do formulário de contato, incluso no módulo em discussão, onde se pode observar algumas das novas

características visuais do MORE.

Figura 19 – Tela do sistema referente ao formulário de contato.



Fonte: print screen da aplicação em execução no navegador *Mozilla Firefox*.

5.1.2 Módulo de Controle de Referências

No módulo de controle de referências as entregas aconteceram segundo os tipos das fontes de referências bibliográficas especificadas nas normas da Associação Brasileira de Normas Técnicas (ABNT).

Estes tipos de fontes são as monografias no todo (livros, dicionários, enciclopédias, relatórios técnicos, anais/proceedings, teses, dissertações e monografias), partes de monografias (capítulos de livros, verbetes de dicionários/enciclopédias e trabalhos apresentados em congressos), periódicos no todo (revistas e jornais), artigos de periódicos (artigos de revistas e artigos de jornais) e documentos exclusivos em meio eletrônico (homepage e e-mail).

A primeira entrega deste módulo referiu-se a implementação das funcionalidades referentes a inclusão e edição de referências a monografias no todo e as demais entregas acontecerem de maneira semelhante para cada tipo de fonte acima citada.

A implementação dos itens de cada tipo de fonte está baseada

em um controlador (contém toda a lógica utilizada para validar os dados vindos dos formulários e construção da página, além de comunicar-se com o modelo e as visões), um modelo (responsável por toda ligação com a base de dados, tanto para consultas como para inserção e atualização) e duas visões (mostram aos usuários um formulário vazio com os campos necessários a elaboração daquele determinado tipo de referência ou, no caso de uma edição, um formulário com os dados disponíveis em disco).

Nesta fase do projeto foram construídos os controladores, modelos e visões para os quinze tipos de referências do sistema legado. A Figura 20 traz a imagem do formulário apresentado ao usuário para a inserção de uma referência a livros.

Figura 20 – Tela do sistema com o formulário de referência a livros.

Fonte: print screen da aplicação em execução no navegador *Mozilla Firefox*.

Para apoiar as alterações necessárias às folhas de estilo que alteram dinamicamente a apresentação dos formulários conforme as escolhas do usuário durante a elaboração de uma referência, assim como as telas *pop-up* de ajuda, criou-se um arquivo JavaScript que encontra-se no diretório *js*, cujo nome é *funfunctions.js*.

As lógicas repetitivas e de uso geral, desenvolvidas em PHP, foram agrupadas em três arquivos distintos que encontram-se no diretório *application/helpers*. São eles: *formatos.helper.php*, *funcoes.helper.php*

e *procedimentos_helper.php*.

Os *helpers* (ajudantes) são arquivos que reúnem um conjunto de funções estruturadas que executam rotinas usadas na criação de formulários, manipulação de *arrays*, customização das mensagens de erro, etc. No contexto deste sistema, as principais funções dos *helpers* são as de implementação das rotinas de verificação e organização dos nomes dos autores, organização e montagem das referências e das citações, verificação da autenticação do usuário, elaboração das mensagens de interação com o usuário, entre outras.

5.1.3 Módulo de Controle de Usuários

Para a implementação do módulo de controle de usuários foram feitas duas entregas. Na primeira, foram incluídos os requisitos referentes à autenticação, abertura e encerramento da sessão de usuário, cadastro e gerenciamento dos dados do usuário (inclusive recuperação automática de senha e/ou sua alteração) e pesquisa.

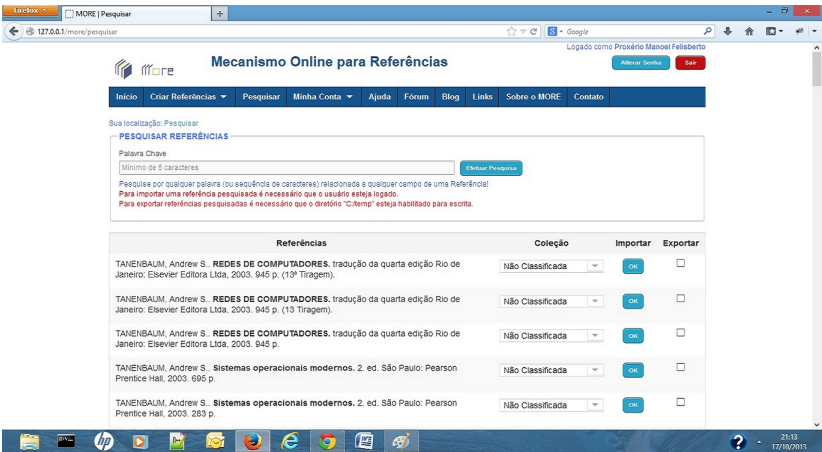
O requisito de pesquisa (sistema de busca), conforme pode ser observado na Figura 21, foi remodelado e reconstruído segundo uma lógica que possibilita as seguintes funcionalidades:

- Encontrar qualquer referência armazenada, informando no mínimo 5 (cinco) caracteres, de qualquer um dos campos de qualquer formulário disponível.
- Após a realização de uma consulta, poderá ir à coluna selecionar e escolher uma ou mais referências e exportá-las para um arquivo *.rtf* que será persistido em disco.
- Se estiver autenticado no sistema poderá utilizar a funcionalidade importar, bastando escolher a qual coleção ficará vinculada. Se nenhuma coleção for selecionada, esta será vinculada à coleção “Não Classificada”. Se o usuário optar por fazer uso desta funcionalidade e não estiver autenticado no sistema, este o direcionará ao formulário de login.

Com a implementação desta lógica se pretende racionalizar a quantidade de dados persistidos em disco, uma vez que não será mais necessário que ocorra duplicação de referências armazenadas, pois uma referência pode estar vinculada a vários usuários, e até mesmo, vinculada a várias coleções.

Na segunda entrega referente a este módulo foram acrescentadas as funcionalidades de inserção e gerenciamento de coleções, bem como

Figura 21 – Tela do sistema com os dados de uma pesquisa.



Fonte: print screen da aplicação em execução no navegador *Mozilla Firefox*.

o gerenciamento de referências. Para que esta etapa fosse levada a efeito, houve a necessidade de adequar a base de dados com adição do campo *CHAVEPROJETO* à entidade *usuarioreferencia* e a criação da entidade *projeto* para possibilitar o gerenciamento de coleções de referências por parte dos usuários finais.

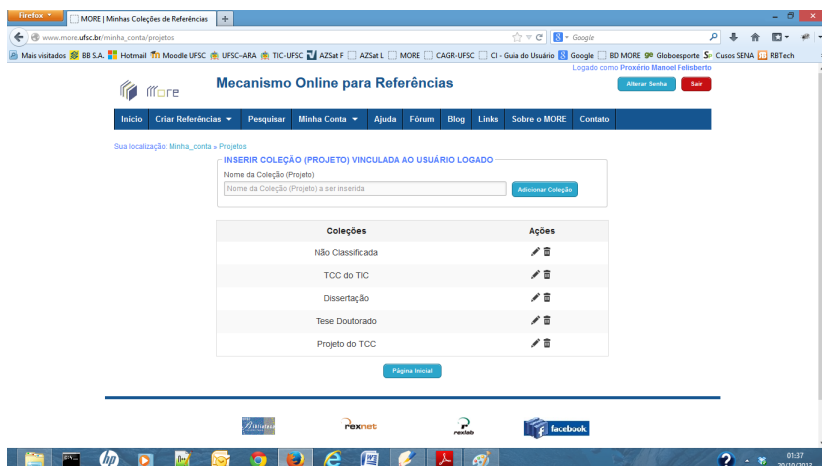
Na mesma página, como pode ser verificado na Figura 22, estando autenticado, o usuário poderá inserir uma coleção que poderá ser usada no gerenciamento de referências, além de editar ou excluir uma coleção previamente cadastrada.

O gerenciamento de referências foi implementado visando oferecer uma interface mais simples e usual. Ao usuário será mostrado suas referências de acordo com a coleção a qual estão vinculadas.

As referências referentes a uma determinada coleção são carregadas em uma tabela na margem direita onde cada linha contém uma referência e os atalhos para edição e exclusão da mesma, como pode ser visto na Figura 23.

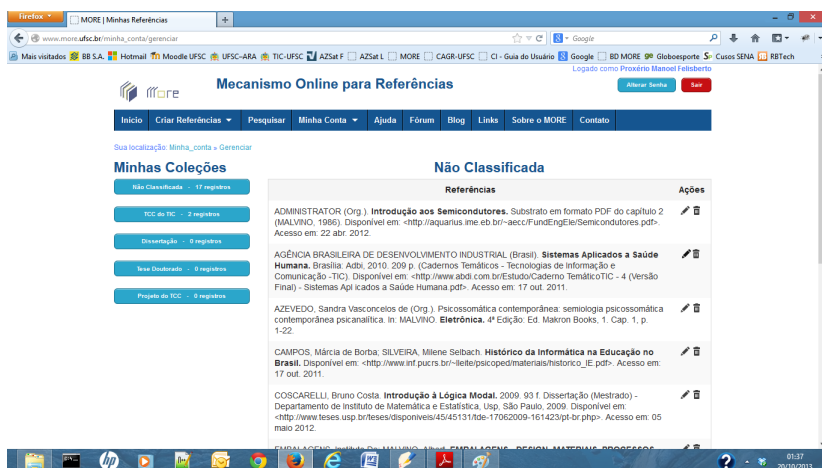
Ao optar por excluir uma referência, será mostrada ao usuário uma tela de confirmação para que se evite uma exclusão indesejada. Ao optar por editar uma referência, esta será localizada na entidade correspondente e seus dados serão passados ao respectivo formulário (que é selecionado automaticamente, através do campo *CHAVETIPO*)

Figura 22 – Tela de inclusão e gerenciamento de coleções.



Fonte: print screen da aplicação em execução no navegador *Mozilla Firefox*.

Figura 23 – Tela de gerenciamento de referências.



Fonte: print screen da aplicação em execução no navegador *Mozilla Firefox*.

que, ao ser apresentado ao usuário, estará com seus campos preenchidos com os dados antigos.

Na margem esquerda encontra-se uma espécie de menu onde cada botão informa o nome da coleção e a quantidade de referências vinculadas a ela. Estes botões constituem-se de enlace para aquela determinada coleção.

5.2 AMPLIAÇÃO DO PORTFÓLIO DE RECURSOS DO SISTEMA

A ampliação do portfólio de recursos do sistema está calcada em uma criteriosa análise realizada nas mensagens de *feedback* dos usuários, recebidas durante os anos de uso do sistema legado e nas solicitações dos usuários da Biblioteca Central desta IES.

Inicialmente, optou-se por construir a lógica e os formulários que contemplassem os documentos jurídicos especificados na NBR 6023 (2002), ou seja, legislação, jurisprudência (decisões judiciais), doutrina e documento jurídico em meio eletrônico. Felizmente, como a estrutura do sistema encontra-se mais flexível e possibilita sua ampliação sem aumentar, de forma perceptível, o nível de complexidade, implementou-se, também, a possibilidade de elaboração de referências a patentes.

Nesta fase foram elicitados mais nove requisitos funcionais que foram inseridos no módulo de controle de referências, seguem o padrão anteriormente descrito e estão especificados abaixo:

- [RF059] *Cadastrar Referência Legislação*: o sistema deverá possibilitar o cadastro de referência a legislação. (Prioridade: Essencial).
- [RF060] *Alterar Referência Legislação*: o sistema deverá permitir a alteração dos dados de uma referência a legislação. (Prioridade: Essencial).
- [RF061] *Excluir Referência Legislação*: o sistema deverá permitir a exclusão de uma referência a legislação. (Prioridade: Essencial).
- [RF062] *Cadastrar Referência Jurisprudência*: o sistema deverá possibilitar o cadastro de referência a jurisprudência. (Prioridade: Essencial).
- [RF063] *Alterar Referência Jurisprudência*: o sistema deverá permitir a alteração dos dados de uma referência a jurisprudência. (Prioridade: Essencial).

- [RF064] *Excluir Referência Jurisprudência*: o sistema deverá permitir a exclusão de uma referência a jurisprudência. (Prioridade: Essencial).
- [RF065] *Cadastrar Referência Patente*: o sistema deverá possibilitar o cadastro de referência a patente. (Prioridade: Essencial).
- [RF066] *Alterar Referência Patente*: o sistema deverá permitir a alteração dos dados de uma referência a patente. (Prioridade: Essencial).
- [RF067] *Excluir Referência Patente*: o sistema deverá permitir a exclusão de uma referência a patente. (Prioridade: Essencial).

O documento de requisitos foi atualizado a fim de comportar e descrever os novos RF segundo o padrão utilizado naquele documento. Não houve alteração dos RNF.

Na Figura 24 observa-se o formulário para a inserção de uma referência a patente, o qual foi implementado no âmbito da ampliação do portfólio de recursos do MORE.

Figura 24 – Tela de elaboração de referência a patente.

The screenshot displays the 'Mecanismo Online para Referências' web application. The browser window shows the URL 'www.more.ufsc.br/patente/insira_patente'. The application has a navigation bar with links like 'Início', 'Criar Referências', 'Pesquisar', 'Minha Conta', 'Ajuda', 'Fórum', 'Blog', 'Links', 'Sobre o MORE', and 'Contato'. The main content area is titled 'ELABORAR REFERÊNCIA A PATENTE' and contains a form with the following fields:

- Escolha uma Coleção de Referências**: A dropdown menu with 'Não Classificada' selected.
- Entidade Responsável**: A text input field with 'Ex: USP/USP' as an example.
- Local (Cidade ou País) ***: A text input field with 'Ex: Brasil' as an example.
- Órgão/Departamento/Repartição**: A text input field with 'Ex: Escola Superior de Agronomia' as an example.
- 1º Autor (entidade criou autor são campos obrigatórios)**: A text input field with 'Nome completo do 1º autor' as an example.
- 2º Autor**: A text input field with 'Nome completo do 2º autor, se houver' as an example.
- 3º Autor**: A text input field with 'Nome completo do 3º autor, se houver' as an example.
- Título ***: A text input field with 'Ex: Medidor digital multi-sensor de temperatura' as an example.
- Sigla do País Emissor ***: A text input field with 'Ex: BR' as an example.
- Número da Patente ***: A text input field with 'Ex: PI 8903105-9' as an example.
- Data do Depósito ***: A date picker with fields for Day (01-31), Month (mm), and Year (aaaa).
- Data da Publicação do Pedido de Patente ***: A date picker with fields for Day (01-31), Month (mm), and Year (aaaa).
- Nome da Publicação (Periódico ou Base de Dados)**: A text input field with 'Ex: Revista Brasileira de Agronomia' as an example.
- Volume**: A text input field with 'Ex: 4' as an example.
- Número**: A text input field with 'Ex: 274' as an example.
- Página Inicial**: A text input field with 'Ex: 74' as an example.
- Página Final**: A text input field with 'Ex: 97' as an example.
- Ano da Publicação**: A text input field with 'Ex: 2012' as an example.
- Informações Complementares**: A text input field with 'Ex: Abstrat 192075a' as an example.
- On-Line?**: A dropdown menu with 'Não' selected.

At the bottom of the form, there are buttons for 'Cancelar' and 'Gravar Referência'. Below the buttons, a message states: 'Seu status é Anônimo, para que você possa gravar suas referências clique no menu Minha Conta ou clique aqui.' and 'Os campos indicados com * são campos de preenchimento obrigatório.'

Fonte: print screen da aplicação em execução no navegador *Mozilla Firefox*.

Vale esclarecer que os documentos jurídicos em meio eletrônico estão contidos nos formulários dos requisitos elicitados e para o usuário

dispor desta possibilidade basta escolher a opção disponível em cada formulário e preencher os dados referentes a *Uniform Resource Locator* (URL) e a data de acesso ao recurso. Os documentos referentes a doutrina se confundem (estão contidos) com outros tipos de fonte. Por exemplo, a doutrina de Damázio provavelmente estará grafada em um livro ou em alguma publicação especializada.

No dia 19 de outubro do corrente ano o MORE foi posto em produção no RexLab Araranguá e seu acesso está disponível através da URL <<http://www.more.ufsc.br>>.

Salienta-se que todo o sistema possui proteção contra a injeção de códigos JavaScript (ou qualquer outro tipo de script) em seus campos de formulário, impedindo que estes códigos cheguem a outros usuários. Esta prática é conhecida como *cross-site scripting* (XSS).

Outra proteção implementada foi contra *cross-site request forgery* (CSRF), ou seja, falsificação de solicitação entre sites. Assim, os redirecionamentos de “sites fantasmas” são bloqueados e não são processados pelo sistema, impedindo o bombardeamento do site. Esta técnica é implementada com um *token* que é gravado na sessão do usuário e alterado a cada mudança de página, não havendo repetição de valores. Se isto ocorrer é prenúncio de que algum redirecionamento malicioso está ocorrendo e portanto será bloqueado.

6 CONSIDERAÇÕES FINAIS E PROPOSTAS PARA TRABALHOS FUTUROS

As tecnologias emergentes para desenvolvimento web, aliadas a melhoria da largura e da qualidade da banda que aumentam a confiabilidade das redes, oferecem uma vasta gama de possibilidades, a maioria delas não imagináveis no passado recente.

Essas possibilidades trazem consigo exigências que a maioria dos softwares legados não são capazes de oferecer, tais como aumento verticalizado da escalabilidade e das possibilidades que devem ofertar aos seus usuários finais, traduzidas numa razão direta em complexibilidade.

Uma vez que o MORE não se caracteriza como exceção, houve a necessidade de sua adequação ao contexto atual. Como não se dispõe de recursos específicos para a manutenção deste empreendimento, buscou-se tecnologias gratuitas que suprissem esta necessidade, porém que garantissem suporte ao desenvolvimento do trabalho na direção dos objetivos almejados.

Após a realização de um estudo e posterior análise sobre o estado da arte, no tocante as tecnologias mais indicadas para o desenvolvimento de sistemas web interativos, iniciou-se o processo de reengenharia do MORE, segundo orienta Sommerville (2011).

Com o emprego da IDE *Astah Community* foi possível fazer a modelagem dos diagramas UML que foram utilizados na elaboração do documento de requisitos. Este documento foi elaborado conforme o modelo adotado na disciplina de Engenharia de Software desta IES e é o local onde constam todas as informações vitais para o entendimento e desenvolvimento do novo sistema.

Assim, o MORE foi reestruturado em três camadas e remodelado a fim de cumprir todos os requisitos elicitados, funcionais e não funcionais. Destaca-se a utilização de dois *frameworks*: o *CodeIgniter* (para dar suporte à reestrutura) e o *Zurb Foundation* (para dar suporte a remodelagem e repaginação visual).

A abordagem realizada na Subseção 5.1.3 e mostrada na Figura 21 caracteriza o desenvolvimento e implementação de um sistema de busca que encontre qualquer referência armazenada, informando no mínimo 5 (cinco) caracteres de qualquer um dos campos de qualquer formulário armazenado, bem como a implementação da possibilidade de importação de uma referência armazenada na base de dados para uma coleção pessoal.

Conforme elucidado na Subseção 5.1.3 e visto nas Figuras 22

e 23 encontra-se disponível ao usuário a possibilidade de cadastrar e gerenciar mais de uma coleção, às quais serão vinculadas as referências que desejar.

No tocante a ampliação do portfólio de recursos do MORE, a abordagem feita na Seção 5.2 e materializada na Figura 24 permite inferir que os resultados alcançados ultrapassaram as expectativas iniciais para este trabalho. Este resultado atribui-se ao fato desta nova versão do sistema possuir uma estrutura mais flexível e robusta, com baixo grau de dependência entre as classes e entre as camadas, facilitando a implementação de novas funcionalidades.

Finalmente, conforme podemos conferir na Seção 5.2, com a entrada em produção do sistema, através de servidor web localizado no RexLab Araranguá, está sendo ofertado aos usuários de biblioteca, que não disponham de uma ferramenta proprietária, a possibilidade de fazer uso de um gerador/gerenciador de referências bibliográficas gratuitamente.

Do exposto até o presente momento e considerando as informações constantes dos Capítulos 5 e 6, deduz-se, com convicção, que o objetivo geral proposto no Capítulo 1, em que se buscava com este trabalho *“Realizar uma reengenharia e ampliar o portfólio de recursos do Mecanismo Online para Referências”* foi plenamente alcançado.

O acompanhamento das visitas ao sistema através do *Google Analytics* nos revela, com grata surpresa, que o tempo de resposta à requisição da página inicial do sistema gira em torno de 3,54 segundos e entre as páginas do sistema este tempo está aquém de um segundo. Estas informações corroboram a expectativa de cumprimento dos RNF em condições normais de operação, principalmente largura de banda e estabilidade da rede.

Infelizmente a racionalização do tempo disponível aliada a escassez de recursos não permitiram que fossem contempladas todas as boas sugestões, quer sejam as recebidas através do canal existente no sistema ou as identificadas na literatura que trata sobre o estado da arte.

Assim sendo, sugere-se como possibilidade de trabalhos futuros a implementação das seguintes funcionalidades:

- Implementar novos formulários de forma a contemplar a elaboração de referências para os demais tipos de fontes listadas na NBR 6023:2002, quais sejam: imagem em movimento, documento iconográfico e documento iconográfico em meio eletrônico, documento cartográfico e documento cartográfico em meio eletrônico, documento sonoro no todo e documento sonoro em parte, docu-

mento tridimensional, partitura e partitura em meio eletrônico.

- Importar informações de bases de dados externas para facilitar o preenchimento dos formulários do sistema.
- Exportar as coleções de referências dos usuários em formatos compatíveis com programas gerenciadores de referências. Por exemplo, o formato RIS utilizado pelo *EndNote*, ou o formato *bibtex* muito utilizado na comunidade científica.
- Gerar referências em outros formatos além do ABNT, tais como o MLA (*Modern Language Association*), *Vancouver* (formato muito utilizado em publicações médicas) e APA (*American Psychological Association*).
- Integração com outras ferramentas de geração e gerenciamento de referências.
- Enviar referências por e-mail.

REFERÊNCIAS

- ACHOUR, M. et al. *Manual do PHP*. 2013. Disponível em: <http://php.net/manual/pt_BR/index.php>. Acessado em 07 maio 2013.
- ALVES, M. B. M.; MENDES, L. L.; ALVES, J. B. da M. More: Mecanismo on-line para referências. 2007. Disponível em: <<http://xa.yimg.com/kq/groups/25169972/1250144979/name/4.ALVES>>. Acessado em 14 ago. 2014.
- AMARAL, L. G. *CSS – Cascating Style Sheets: guia de consulta rápida*. 3. ed. São Paulo: Novatec Editora, 2009. 120 p.
- ANTHES, G. HTML5 leads a web revolution. *Communications of the ACM*, v. 55, n. 7, p. 16–17, July 2012.
- APPCCELERATOR. *Documentation*. 2012. Disponível em: <<https://wiki.appcelerator.org/display/tis/Home>>. Acessado em 28 dez. 2012.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 10520: Informação e Documentação - Citações em Documentos - Apresentação*. Rio de Janeiro, ago. 2002. 7 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 6023: Informação e Documentação - Referências - Elaboração*. Rio de Janeiro, ago. 2002. 24 p.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *Conheça a ABNT*. 2013. Disponível em: <http://www.abnt.org.br/m3.asp?cod_pagina=929>. Acessado em 21 out. 2013.
- BLANKENSHIP, J.; BUSSA, M.; MILLETT, S. *Pro Agile .NET Development with Scrum*. New York, US: Apress Media, 2012. 381 p.
- BÖCK, H. *The Definitive Guide Netbeans Platform 7*. New York, US: Apress, 2012. 529 p.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. Rio de Janeiro: Elsevier, 2012. 521 p. Tradução de Fábio Freitas da Silva e Cristina de Amorim Machado. 12ª reimpressão.

BRONDANI, C. H. et al. Guia prático de utilização da ferramenta astah community 6.1. 2011. Disponível em: <<http://www-pet-si.inf.ufsm.br/images/consultoriodesoftware/Astah.pdf>>. Acessado em 27 set. 2013.

CERVO, A. L.; BERVIAN, P. A.; SILVA, R. da. *Metodologia Científica*. 6. ed. São Paulo: Pearson Prentice Hall, 2007. 159 p.

CHANGE VISION. *Astah: Basic operation guide*. [S.l.], 2009. 50 p. Disponível em: <<http://astah.net/resources/documents/astah-basic-operation.pdf>>. Acessado em 25 set. 2013.

CHANGE VISION. *Astah Reference Manual*. [S.l.], 2012. 433 p. Disponível em: <<http://astah.net/tutorials/astah>>Acessado em 27 set. 2013.

CINAR, O. *Android Apps with Eclipse*. New York, US: Apress Media, 2012. 358 p.

DANTAS, R. *NetBeans IDE 7 Cookbook*. Birmingham, UK: Packt Publishing, 2011. 290 p.

DEULING, T. *Aptana Studio Beginner's Guide*. Birmingham, UK: Packt Publishing, 2013. 283 p.

E-PRINTS. *Guía ProCite® 5.0*. 2006. Disponível em: <<http://eprints.rclis.org/8112/1/ProCite50.pdf>>. Acessado em 07 set. 2013.

ECLIPSE FOUNDATION. *PDT Documents*. 2013. Disponível em: <<http://www.eclipse.org/pdt/docs.php>>. Acessado em 21 set. 2013.

ELLISLAB. *CodeIgniter User Guide*. [S.l.], 2013. Disponível em: <<http://ellislab.com/codeigniter/userguide/>>. Acessado em 17 abr. 2013.

ESTORNIOLO FILHO, J. *EndNote Web: guia de uso*. 6. ed. São Paulo, jan. 2013. 29 p. Disponível em: <<http://www.bvs-sp.fsp.usp.br:8080/image/pt/internas/manuais/endnoteweb.pdf>>. Acessado em 06 set. 2013.

GABARDO, A. C. *PHP e MVC com CodeIgniter*. 1. ed. São Paulo: Novatec Editora, 2012. 286 p.

GILMORE, W. J. *Dominando PHP e MySQL: do iniciante ao profissional*. 3. ed. Rio de Janeiro: Alta Books, 2011. 769 p. Tradução de Raquel Marques e Lúcia Kinoshita. 2ª reimpressão.

GONÇALVES, E. *Dominando Netbeans*. Rio de Janeiro: Ciência Moderna, 2006. 363 p.

HOLZNER, S. *Eclipse*. Sebastopol, US: O'Reilly Media, 2004. 317 p.

ISI RESEARCHSOFT. *ProCite®: User's guide*. Berkeley - USA, 1999.

LOUDON, K. *Desenvolvimento de Grandes Aplicações Web*. São Paulo: Novatec Editora, 2010. Tradução: Rafael Zanolli.

MACINTYRE, P. B. *PHP: The good parts*. 1. ed. Sebastopol, USA: O'Reilly Media, 2010. 158 p.

NETCRAFT. *PHP just grows & grows*. 2013. Disponível em: <<http://news.netcraft.com/archives/2013/01/31/phpjustgrows-grows.html>>. Acessado em 02 out. 2013.

NIEDERAUER, J. *Web Interativa com Ajax e PHP*. São Paulo: Novatec Editora, 2007. 288 p.

PALOMAR COLLEGE LIBRARY. *EasyBib: Write Smart*. [S.l.], 2013. Disponível em: <<http://www.palomar.edu/library/easybib/what-is-EasyBib.pdf>>. Acessado em 15 ago. 2013.

PESTANA, M. C.; FUNARO, V. M. de O. *Manual EndNote Web: guia do usuário*. São Paulo, 2009. 23 p. Disponível em: <http://143.107.23.244/sdo/arquivos/Manual_EndNote_Web.pdf>. Acessado em 06 set. 2013.

PINTO, F. de S.; RIBEIRO, L. F. *Facilis: Ferramenta on-line para geração de referências de acordo com as normas da ABNT*. 2010. Disponível em: <<http://www.cesmac.com.br/erbase2010/papers/wtictg/65789.pdf>>. Acessado em 13 ago. 2013.

POWERS, S. *Aprendendo JavaScript*. São Paulo: Novatec Editora, 2010. 407 p. Tradução: Acauan Pereira Fernandes.

PRESSMAN, R. S. *Engenharia de Software: uma abordagem profissional*. 7. ed. Porto Alegre: AMGH, 2011. 771 p. Tradução: Ariovaldo Griesi.

PROQUEST. *Guia de início rápido de Refworks 2.0*. 2013. Disponível em: <https://www.refworks.com/refworks2/help/RefWorks_QSG_ES_Dec11.pdf>. Acessado em 16 set. 2013.

PROQUEST. *Help File RefWorks*. 2013. Disponível em: <<http://www.refworks.com/refworks2/help/RefWorks2.htm>>. Acessado em 15 set. 2013.

REBITTE, L.; VINICIUS, M. *Dominando Tableless: seu site entre os primeiros nos sites de busca!* Rio de Janeiro: Alta, 2006. 115 p.

SÁLVIO, S. C. *GUIA DE UTILIZAÇÃO DO ENDNOTE WEB*. São Paulo, 2013. Disponível em: <<http://www.sbu.unicamp.br/fontes-eletronicas/images/PDF/endnote.pdf>>. Acessado em 25 jul. 2013.

SCHWARTZ, B. et al. *High Performance MySQL*. Second edition. Sebastopol, USA: O'Reilly Media, 2008. 684 p.

SERRA, A. P. G. Reengenharia de software: Uma visão geral. *Engenharia de Software Magazine*, Rio de Janeiro, n. 11, p. 20–24, 2009. Mensal.

SILVA, M. S. *JavaScript: guia do programador*. São Paulo: Novatec Editora, 2010. 604 p.

SILVA, M. S. *HTML 5: a linguagem de marcação que revolucionou a web*. São Paulo: Novatec Editora, 2011. 320 p.

SILVA, M. S. *Ajax com jQuery: requisições ajax com a simplicidade do jquery*. São Paulo: Novatec Editora, 2012. 327 p. Terceira reimpressão.

SILVA, M. S. *CSS3: desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das css3*. São Paulo: Novatec Editora, 2012.

SOMMERVILLE, I. *Engenharia de Software*. 9. ed. São Paulo: Pearson Prentice Hall, 2011. 529 p. Tradução Ivan Bosnic e Kalinga G. de O. Gonçalves.

SOURCEFORGE.NET. *JabRef Reference Manager: documentation*. 2013. Disponível em: <<http://jabref.sourceforge.net/documentation.php>>. Acessado em 05 set. 2013.

THE LANDMARK PROJECT. *Citation Machine*. [S.l.], 2013. Disponível em: <<http://citationmachine.net/index2.php>>. Acessado em 15 ago. 2013.

THOMSON REUTERS. *Reference Manager: Getting started guide*. [S.l.], 2008. 80 p. Disponível em: <<http://www.refman.com/support/docs/rmgettingstartedguide12.pdf>>. Acessado em 09 set. 2013.

THOMSON REUTERS. *Reference Manager: Product Information*. [S.l.], 2013. Disponível em: <<http://refman.com/rminfo.asp>>. Acessado em 15 ago. 2013.

TONU, M. A. H. *PHP Application Development with NetBeans: Beginner's guide*. Birmingham, UK: Packt Publishing, 2012. 283 p.

UNIVERSIDAD AUTÓNOMA DE MADRID. *RefWorks: Manual de usuario*. Madrid, 2008. 28 p. Disponível em: <<http://biblioteca.uam.es/sc/documentos/refworks.pdf>>. Acessado em 12 set. 2013.

UNIVERSIDAD DE SALAMANCA. *RefShare: Manual de usuario*. Salamanca, 2006. Disponível em: <http://sabus.usal.es/pdf/manual_share.pdf>. Acessado em 17 set. 2013.

UNIVERSIDAD DE SEVILLA. *PROCITE: Gestor de referencias bibliográficas*. fev. 2006. Disponível em: <http://bib.us.es/aprendizaje_investigacion/publicar_citar/herramientas/common/procite.pdf>. Acessado em 09 set. 2013.

UNIVERSIDAD NACIONAL DE COLOMBIA. *Reference Manager: Guía de usuario*. Santafé de Bogotá, D.C., 2006. 62 p. Disponível em: <<http://www.unalmed.edu.co/befego/documentos/Tutoriales/ReferenceManage.pdf>>. Acessado em 09 set. 2013.

UNIVERSIDAD REY JUAN CARLOS. *RefWorks: Manual de uso*. Madrid, 2007. 35 p. Disponível em: <http://www.urjc.es/biblioteca/Archivos/manual_refworks.pdf>. Acessado em 17 set. 2013.

UPTON, D. *CodeIgniter for Rapid PHP Application Development*. Birmingham, UK: Packt Publishing, 2007. 244 p.

VAUGHAN-NICHOLS, S. J. Will HTML5 restandardize the web? *Computer*, v. 43, n. 4, p. 13–15, April 2010.

ZURB. *Foundation 3 Documentation*. 2013. Disponível em: <<http://foundation.zurb.com/old-docs/f3/>>. Acessado em 07 out. 2013.